

CP-PACSパイロットモデルにおけるNAS並列ベンチマークの評価

服部 正樹 板倉 憲一 朴 泰祐 中村 宏 中澤 喜三郎

筑波大学 電子・情報工学系
〒305 つくば市天王台1-1-1

Tel: 0298-53-6912 Fax: 0298-53-5206

E-mail: {hattori,itakura,taisuke,nakamura,nakazawa}@arch.is.tsukuba.ac.jp

あらまし

本研究では、大規模科学技術計算を目的とした超並列計算機 CP-PACS のプロトタイプであるパイロットモデル (PILOT-1) の性能評価を実測によって行なう。PILOT-1 はノードプロセッサに PA-RISC1.1 アーキテクチャのスーパースカラプロセッサを用い、それらをハイバクロスバ・ネットワークで結合した分散メモリ型並列計算機である。

評価はNAS並列ベンチマークを用いて行なった。評価結果から、PILOT-1 は各ベンチマークによって求められる様々な転送パターンに対して柔軟に対応でき、様々な問題に対して最適なPU間通信が行なえることが分かった。また、十分な粒度を保っている問題ではほぼ線形な台数効果が得られた。

NAS Parallel Benchmarks Evaluation on CP-PACS Pilot-Model

Masaki HATTORI Ken'ichi ITAKURA Taisuke BOKU

Hiroshi NAKAMURA Kisaburo NAKAZAWA

Institute of Information Sciences and Electronics

University of Tsukuba

1-1-1 Tenmodai, Tsukuba 305

Tel: 0298-53-6912 Fax: 0298-53-5206

E-mail: {hattori,itakura,taisuke,nakamura,nakazawa}@arch.is.tsukuba.ac.jp

Abstract

In this paper, we evaluate the total performance of the prototype of CP-PACS named PILOT-1 by actual measurement. CP-PACS, a massively parallel processor, is aimed to solve large scale scientific problems. PILOT-1 is a parallel processor with distributed memory architecture, and equipped with superscalar processors based on PA-RISC 1.1 architecture as nodes and Hyper-Crossbar Network as interconnection network among nodes.

We evaluate the performance of PILOT-1 on NAS Parallel Benchmarks. As a result of evaluation, PILOT-1 can be applied flexibly to various patterns of data transferring, and so provides the most suitable way of communication among nodes. PILOT-1 also achieves a linear speedup in some kernels which can keep enough granularity.

1 はじめに

本研究では、超並列計算機 CP-PACS (Computational Physics by Parallel Array Computer System) [1] のプロトタイプである CP-PACS PILOT-1 (以下 PILOT-1 と略) の性能を、実測に基づき評価する。

CP-PACS は現在筑波大学を中心に開発が進められており、計算物理学の分野における大規模科学技術計算を主たる目的とする。CP-PACS は PU (Processing Unit) 間を HXB (Hyper-Crossbar Network) [2] で結合した分散メモリ型超並列計算機である。HXB は柔軟性が非常に高く、転送性能を低下させることなく多様な PU 空間を提供することができる [3]。PILOT-1 は同様に HXB を実装した分散メモリ型並列計算機で、同大学の計算物理学研究センターに設置されており、現在稼働中である。PILOT-1 上で各種問題を実測によって評価することにより、実機の CP-PACS の性能を予測することができると思われる。

本研究では、PILOT-1 の性能評価に NAS 並列ベンチマーク [4] のカーネル問題を用いる。これらの問題は基本的な解法が与えられているが、並列化の手法は各並列計算機に適したものを考えて良い。PILOT-1 においても個々の問題に対して、HXB の柔軟性を考慮し、データのマッピング、転送パターンなどの点から最適な並列化について考察する。

2 CP-PACS PILOT-1

まず、PILOT-1 のシステム概要について説明する。PILOT-1 は MIMD 型並列計算機で、14 台の PU と 2 台の IOU (うち 1 台は SIOU)、計 16 台のノードから構成されている (図 1)。IOU (Input/Output Unit) は分散磁気ディスク記憶装置などを接続するためのアダプターが備わっている PU であり、SIOU (Supervisor Input/Output Unit) はコンソールが接続された IOU である。相互結合網には 2 次元 HXB を用いており、その最大スループットは 100MB/s である。ただし、NIA (Network Interface Adapter) とメモリ間のシステムバスの最大スループットが 80MB/s となっており、実質的な最大スループットは 80MB/s となっている。

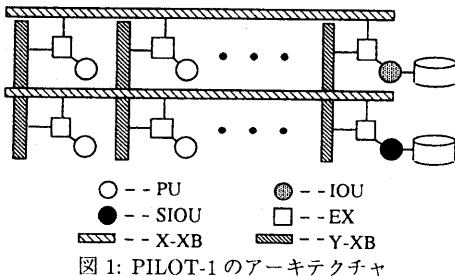


図 1: PILOT-1 のアーキテクチャ

ノードプロセッサには HP 社の PA-RISC1.1 アーキテクチャを採用した RISC プロセッサを使用している。

2 命令並列実行可能な superscalar 方式で、クロック周波数は 90MHz、主記憶容量は 32MB (IOU は 64MB) となっている。ストアイン・ダイレクトマップ方式の 2 レベルキャッシュを用いており、その容量は 1 次が命令 8KB / データ 16KB、2 次が命令 1MB / データ 1MB となっている。

PILOT-1 上では C 言語及び Fortran-77 言語によるプログラミングが可能である。これらの言語に並列プロセス生成・プロセス間通信用のライブラリ関数が追加されており、それらを用いて明示的にメッセージパッシングを行なう並列プログラムを作成する。

PILOT-1 で用いる 2 次元 HXB は、2 次元正方格子状にノードを並べてその間をクロスバスイッチ (XB) で結合した間接網である。各次元方向のノードは XB によって完全結合され、それ以外のノード間のデータ転送はエクステンジャー (EX) と呼ばれる小型のクロスバスイッチによって複数次元方向の XB を経由して行なう。PILOT-1 ではメッセージ転送に wormhole ルーティングを用い、経路の選択はデッドロックを防ぐために固定ルーティング方式を用いる。HXB は基本転送に関するコストパフォーマンスが良い、ランダム転送時の性能が高いといった利点に加え、別構成の HXB やいくつかの別トポロジのネットワークのエミュレートが可能であるといった特徴を持っている。更に、2 次元 HXB では論理的 2 次元 PU 空間をそのまま問題空間にマッピングして用いたり、1 次元 PU 空間に展開して用いることができるといった柔軟性を持っている [3]。

PILOT-1 のノード間通信は、リモート DMA 転送と呼ばれる。これは、送信側・受信側共にユーザの仮想アドレス空間の一部を物理メモリ上に固定的に割り当てておき、それらのメモリ間でデータ転送を行う高速通信方式である。異なったノード上の、ユーザ空間同士で直接データ転送を行うため、カーネルとユーザ空間との間でのデータコピーが発生せず、ネットワークのスループットを最大限に発揮することができる。この機能は転送データのサイズが大きくなるほど有利に働く。アプリケーション上の使用方法としては、計算領域を直接通信領域として使う方法が最も理想的である。

また、リモート DMA 転送では NIA の起動をごく軽いシステムコールによって行ない、通信立ち上げオーバーヘッドを軽減するモードが用意されている。これは小サイズのデータを何度も転送する場合に有利となる。

リモート DMA 転送は高速ではあるが、使い方に注意が必要である。最も重要な点は送信側が受信側のメモリに直接書き込むため、書き込みのタイミングを間違えると有用なデータ上にオーバーライトしてしまうという問題である。更に、キャッシュとメモリの整合性の問題もあるが、今回の評価ではメッセージ送出時及び到着時に、それぞれのノードで OS によってソフトウェア的にキャッシュとメモリの整合性をとる、というモードで通信を行なわせることによりこれを解決している。また、リモート DMA 転送では通信領域は物理メモリ

に割り当てられた連続領域上になければならないという制約がある。

3 NAS 並列ベンチマーク

NAS 並列ベンチマーク [4](以下 NPB) は、科学技術計算を中心とした並列計算機のためのベンチマーク問題であり、5つのカーネルプログラムと3つの応用プログラムから構成される。本研究ではそのうちのISを除く4つのカーネル(表1)について評価を行なう。なお、全ての浮動少数点データは倍精度となっている。

表 1: NPB のカーネルプログラム

EP	乗算合同法による乱数生成
MG	multi-grid 法によるポアソン方程式の求解
CG	CG 法による大規模疎行列の最小固有値問題
FT	多次元 FFT による 3 次元偏微分方程式の求解

NPB の特徴は、問題がソースプログラムではなく、アルゴリズムによって与えられる点である。ここでは問題の規模、初期データ、解法のアルゴリズムのみが与えられており、各計算機のための並列化はユーザが自由に行なうことにより、その計算機に適した方法で性能を評価することができる [5]。

NPB カーネルには、問題規模に応じて tiny、Class A、Class B の 3 種類が存在する。本研究では PILOT-1 のノード数を考慮し、tiny 及び Class A を対象とする。

4 各カーネルの評価及び考察

PILOT-1 上で動作する各カーネルの並列プログラムを、Fortran 言語と通信ライブラリを用いて作成する。コンパイラの最適化オプションとして、最高レベルの最適化、ループアンローリング、外部手続きインライン展開、ソフトウェアパイプライン化の4つを指定する。

スピードアップ率の計算は1ノードの実行時間を基準に行なうが、これは通信を行わず逐次に実行するプログラムを並列プログラムとは別に作成し、IOU、SIOU以外のノード上で実行させた結果である。また、14ノードまではIOUとSIOUを、15ノードまではSIOUを除いたノード上で実行する。

通信時間の見積りを計算するために、あらかじめリモートDMA転送による単純な転送実験を行なって、OSによるキャッシュの制御を伴う時の転送のスループットと立ち上げオーバーヘッドを実測した。結果は、スループット 36MB/s、立ち上げオーバーヘッド 100 μ sec となった。よって $N\frac{1}{2}$ (最大転送性能の半分の性能を出すメッセージ長) は約 3.7KB となる。

指定された範囲の elapse 時間と CPU 処理時間の測定を行ない、その差が通信時間であると考えて、CPU 時間と通信時間の割合を算出する。

なお、各カーネルの結果を示す表中の #PU はノード数、EXEC は全実行時間 (sec)、TRANS は全通信時間 (sec)、RATIO は全実行時間に占める通信時間の割

合 (%)、PERF は実効計算性能 (MFLOPS) をそれぞれ表している。

4.1 EP (完全並列問題)

EP はモンテカルロ・シミュレーションの典型例であり、ほぼ完全な並列化ができる。これによりノード単体の浮動少数点演算性能の実効的な上限が求められる。

EP は tiny、Class A でそれぞれ 2^{21} 、 2^{28} 組の乱数に対して処理を行なう。1次元の問題空間に対してノードを1次元状に展開してマッピングし、各々のノードで問題を完全に並列化して処理を行なう。各ノードで計算した結果を1つのノードに集めて集計し、最終的な結果を求める。通信はこの時にだけ必要となる。

各サイズの結果を表2及び図2に示す。

表 2: EP の実行時間と性能

#PU	tiny		Class A	
	EXEC	PERF	EXEC	PERF
1	106.664	13.050	1688.140	13.149
2	53.220	26.156	847.675	26.186
4	26.525	52.479	422.236	52.570
8	13.454	103.463	214.294	103.582
16	7.316	190.264	109.540	202.638

EP はデータ領域が小さいためデータキャッシュがほぼオールヒットしていると考えられるが、内部処理に平方根と log の計算が多く含まれており、これらを粗込み関数を用いてソフトウェア的に処理していることが、性能が低くなっている要因であると考えられる。

EP はその処理のほとんどが CPU の内部処理で占められ、全処理時間に対する転送時間の割合が小さいため、ほぼ完全な台数効果が現れている。

0 番のノードは SIOU となっており、これは他の PU とは違い、システム中のフロントエンドプロセスの実行を担っているため、使用可能なメモリ、キャッシュなどの資源が他のノードに比べて少なく、性能も低い。16 台のノードを全て使用すると、この性能の低い SIOU が計算に参加するため、全体の性能が低下することが予想される。図2で tiny の 16 台の時の性能が 15 台の時よりも落ちているのは、これが原因であると思われる。

4.2 MG (multi-grid 法)

MG は V-cycle multi-grid 法による 3 次元ポアソン方程式の求解を行なう問題である。Class A では $256 \times 256 \times 256$ 点、tiny では $32 \times 32 \times 32$ 点の格子点について、ポアソン方程式を周期境界条件の下で解く。

MG は各点に対して、周囲 27 点のデータによる値の更新処理を繰り返す。この結果、各ノードは割り当てられたデータの更新と、隣接するノード間でのデータ転送を繰り返すことになる。PILOT-1 では 1 次元及び 2 次元の PU 空間が考えられるが、後者はストライドデータの転送が必要となり、現在の PILOT-1 の OS で

はストライド転送がサポートされていないので、ここでは1次元空間へのマッピングを行なう。

MGでは問題空間の各次元の両端で、各々反対側の面のデータを持たせるためのバッファを設けることで周期境界性を実現する。1次元空間へのマッピングを行なった場合、切り分けた面にもこのバッファを設け、隣のノードから送られてきた面のデータをここに受信する。従って、切り分けた面とその上のバッファの部分が必要最低限の通信領域となる。ところが、PILOT-1のリモートDMA転送では全ての通信領域は連続領域上になければならないという制約があり、このため1ノードが持つ全データ領域を1つの通信領域としなければならない。現在のPILOT-1ではメモリ不足のためMGに関してはtinyのみを評価する。

MGでは処理の途中で問題空間のサイズが変化する。具体的には問題空間が $(1/2)^3$ ずつ小さくなっていき 2^3 の大きさになった後、今度は 2^3 倍ずつ大きくなり最後に元の大きさになる。例えばノード数が8台の時には問題空間のサイズが 8^3 までは全ノードで行なう並列性があるが、それ以下は並列度がノード数を下回る。この状態で無理に並列処理を進めても処理の粒度が小さく、逆に速度低下を招く危険がある。よってノード数分の並列性がなくなった後は1台のみで行なう。

結果を表3及び図2に示す。

表3: MG(tiny)の実行時間と性能

#PU	EXEC	TRANS	RATIO	PERF
1	0.848	0.000	0.000	15.093
2	0.475	0.015	3.158	26.936
4	0.287	0.037	12.892	44.632
8	0.188	0.018	9.574	67.905

MGは1ノードの時に既に全問題領域がキャッシュに載るため、比較的高い性能を達成している。

CPUの内部処理時間は並列実行部分と逐次実行部分に分けられる。並列部分については問題サイズの変化と端数処理の関係から、ノード数が2の冪乗で増加したときだけ処理時間が半分に減少する。逐次部分については2ノードの時には全て並列に実行でき逐次部分はないが、3~4ノードの時には 4^3 まで、5~8ノードの時には 8^3 まで、9ノード以上の時には 16^3 までしか並列にできず、それ以下のサイズでは逐次実行となる。並列部分と逐次部分のトレードオフにより、全内部処理時間はノード数の逆数に必ずしも比例しない。通信について見ると、1回の最大転送量は2.5KB、転送回数は168回以下とそれほど多くなく、通信時間の割合は多くても13%程度でしかない。

以上のことから、CPUの内部処理の問題によりtinyのような小さいサイズでは台数効果はあまり望めず、また、ノード数が2の冪乗でない時の性能は全くノード数に見合ったものにならないことが考えられる。結果は図2の通り、2の冪乗以外のノード数では台数効果が全く現われていない。

4.3 CG (CG法)

CGは正定値対称な大規模疎行列の最小固有値をCG (Conjugate Gradient) 法により求める問題である。ここで用いる大規模疎行列Aは正定値対称であり乱数を用いて生成する。

CG法で行う演算は行列とベクトルの積、ベクトルの内積、ベクトルとスカラの積和計算であるが、行列とベクトルの積($q = A \cdot p$)が最も重い処理なのでこの部分を中心に並列化を考え、ここでは以下の行分割方式のマッピングを行なう。ノードを1次元に展開し行列Aを行分割し、qもAと同様に分割する。pは担当するAの列要素に対応する行要素だけがあればよいが、基本的に全要素が必要となる。CG法は反復法であり、pはqを元にして作られるので、各ノード内で求めたqの一部を全ノードに転送する、すなわち全対全ブロードキャストを行なう必要がある。全対全ブロードキャストはノードをリング状に並べ、隣へのデータ転送を繰り返すことで実現する。

各サイズの結果を表4、5及び図2に示す。

表4: CG(tiny)の実行時間と性能

#PU	EXEC	TRANS	RATIO	PERF
1	5.473	0.000	0.000	13.814
2	3.980	0.678	17.035	18.994
4	2.619	0.819	31.272	28.871
8	2.597	1.424	54.833	29.114
16	4.292	3.732	86.953	17.616

表5: CG(Class A)の実行時間と性能

#PU	EXEC	TRANS	RATIO	PERF
1	278.980	0.000	0.000	5.520
2	140.386	1.708	1.217	10.970
4	71.853	2.074	2.886	21.433
8	37.878	2.863	7.558	40.657
16	20.975	4.592	21.893	73.422

リストベクトル処理のためにキャッシュが効果的に働かず、全体的に性能が低くなっていると考えられる。

CGで行なう全対全ブロードキャストはノード数Pに対して $O(P)$ の転送回数がかかるので、ノード数が多い時にボトルネックになる。また、ノード数が多くなるほど1回の転送量が少なくなるので、問題サイズが小さい場合には転送の立ち上げオーバーヘッドが大きなボトルネックとなる。つまりCGでは、並列度を上げて粒度を細かくすればするほど転送がボトルネックになって、台数効果が得られなくなると予想できる。

tinyでは1回の転送量が最大でも約5.5KBと小さく、かつCPU内部処理量が小さいため、立ち上げオーバーヘッドの累積が台数効果を大きく妨げている。全対全ブロードキャスト転送はキャッシュの処理を行なう必要がないので、立ち上げオーバーヘッドは約50μsecとなるが、その累積時間だけでも4ノードの時に既に全処理時間の約7%を占める計算になる。一方Class Aでは

1回の転送量は最低でも約6.8KBとなっており、内部処理量が大きいため、16ノードでも立ち上げオーバーヘッドの累積は全処理時間に対して約4%程しかなく、十分な粒度を保てるため良い台数効果が得られている。

4.4 FT (多次元FFT問題)

FTは、高速フーリエ変換(FFT:Fast Fourier Transform)を用いて偏微分方程式を数値的に解く問題である。問題空間は3次元であり、この空間上の点を3次元FFT処理し、ある係数を掛けた後に、3次元逆FFT処理を行なうことによって、任意の時刻の状態を調べることができる。問題空間はClass Aで $256 \times 256 \times 128$ 点、tinyで $64 \times 64 \times 64$ 点の複素数で与えられる。方程式の結果に対し、そのうちのランダムな1024点のデータを集計し、checksumを計算する。

FTにおいては、3次元FFTの処理をどのように並列化して行なうかが問題となる。多次元FFTは、全データに対し1次元FFTを各次元毎に1回ずつ行なう。1次元FFTの処理内容はバタフライ演算とビットリバース演算からなる。

PILOT-1では、2次元のPU空間を1次元に展開し、3次元の問題空間をZ次元方向に対して分割してマッピングする。X及びY次元方向のFFTはノード内で完全に閉じた演算となる。Z次元方向FFTのバタフライ演算は、ノード間の2点でのバタフライ演算とノード内の2点でのバタフライ演算に分けられる。前者はノード間で自分が持つ全データを相互にバタフライ転送を行なって交換する。HXB上ではバタフライ転送を無衝突で行うことが可能である。必要となるバタフライ転送の回数は、ノード数がPの時に $\log_2 P$ 回となる。

FTはノードが持つ全データを転送する必要があり、大量の通信領域を必要とするため、Class Aでは全ノードが、またtinyではSIOUがメモリ不足のため使用できない。更に、バタフライ転送はノード数が2の冪乗に限られるため、8ノード以下でのtinyの評価のみを行なう。

結果を表6及び図2に示す。

表6: FT(tiny)の実行時間と性能

#PU	EXEC	TRANS	RATIO	PERF
1	22.754	0.000	0.000	8.438
2	14.304	0.732	5.117	13.423
4	8.260	0.784	9.491	23.245
8	4.307	0.653	15.161	44.583

バタフライ転送の転送回数はノード数Pに対して $O(\log(P))$ で増加するが、1ノードあたりの転送量は $O(1/P)$ で減少する。更にHXBではこれが無衝突で転送可能である。したがってノード数が増えるほどバタフライ転送の通信時間は減少することになる。しかし、ノード数の増加に伴ってビットリバース転送での衝突が増大するため、全通信時間は単純には減少しない。

メモリ上に連続していない方向の1次元FFTでは、ラインコンフリクトによるキャッシュミスが頻発し、全体的にFLOPSが低い原因となっている。しかしノード数の増加と共にキャッシュが有効に働くようになり、8ノードの時は全データがキャッシュに収まるため、CPU時間はスーパーニア的に減少している。しかし、通信に伴うキャッシュミスが原因となってCPU時間はそれほど著しく減少しない。即ち、転送されてきたデータはメモリ上にしかなく、キャッシュミス率の増大につながる。多くの転送量を必要とするFTではこれがかなり影響しており、台数効果が妨げられていると思われる。

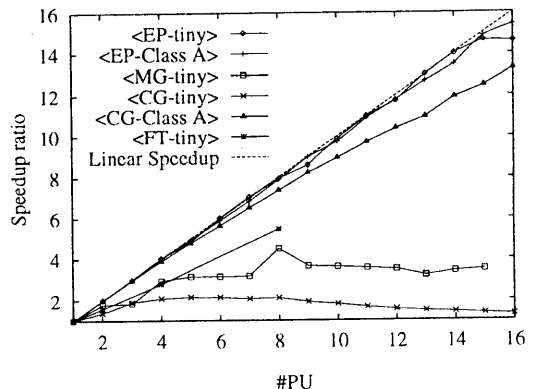


図2: 各カーネルのスピードアップ率

5 総合評価

各カーネルにおける実効性能を図3に示す。

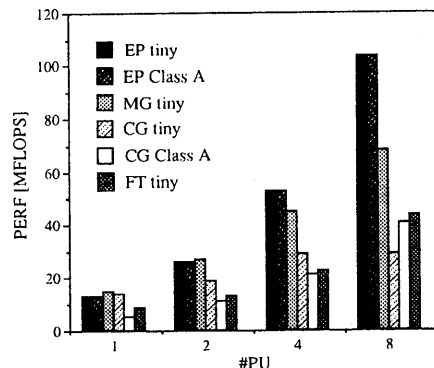


図3: 各カーネルの実効性能

EPは完全並列な問題で、またキャッシュヒット率が高いため、4つのカーネル中最大の性能を出している。tinyとClass Aが同じ性能であることから、並列処理のオーバーヘッドがほぼ1であることが分かる。次に高い性能を示すのはMGで、これもキャッシュが有効に働くことと通信時間の割合が小さいことが影響している。逆にCGのtinyは通信がネックとなり、あまり性能が出ていない。CGのClass Aはノード数が小

さい時はノードあたりの問題サイズが大きいためキャッシュミス率が高く、また転送量が大きいため通信がネックとなることにより性能が低くなっているが、ノード数が増えても十分な粒度を保てるために台数効果が現われ、8ノードでtinyとの逆転が見られる。FTは膨大な転送量が必要となるにも関わらず、問題領域がノード数によって分割されキャッシュが有効に働くことと、HXBにおけるバタフライ転送の有利性と高スループットによって、ある程度の台数効果が得られている。

HXBとリモートDMA転送の有効性によって通信時間を削減できるのだが、メモリ容量の不足のためにMGとFTのClass Aが実行できず、tinyでは粒度が細か過ぎるために大きな台数効果を得られないという問題があった。Class Aが可能であれば十分な台数効果を発揮でき、スーパリア効果を得られる可能性もある。

また、ノードの構成台数にも問題がある。SIOUはその性質上、計算に参加することが不適當であったり、場合によっては不可能であったりする。そうすると事実上、最大並列度は15ということになるが、FTのようにノード数が2の冪乗に限られたり、可能であってもMGのように2の冪乗以外のノード数では台数効果を得られなかったりするため、実効的には最大並列度が8にしかないことがある。実際のアプリケーションでもFTやMGのようなことがあり得るため、純粋なPUのみで2の冪乗台の計算専用PU群を構成し、IOU及びSIOUはそれとは別に設けるべきである。

他の並列計算機と比較した場合、今回の結果に見るPILOT-1の性能は良いとはいえない。これはCPUの性能と転送の立ち上げオーバーヘッド、通信に伴うキャッシュミスに大きく依存していると思われる。PILOT-1に用いられている90MHzのPA-RISC1.1は、現在の最高速度を持つ他の並列機に比べ特に速いものではない。転送の立ち上げオーバーヘッドについては、PILOT-1ではより小さいオーバーヘッドで転送を行なうことも可能だが、現在のFortran言語の通信関数とそのモードをサポートしておらず、使用できなかった。これが解決できれば転送性能の大幅な向上が期待される。

各カーネルには、まだ並列化や内部処理のアルゴリズムにおいて更にチューニングの可能性が有ると思われる。EPはその内部処理において、ソフトウェアパイプライン化を積極的に行なうことにより、更に高い性能を発揮すると予想される。MGについては逐次実行部分を減らす工夫を行なうことで、台数効果を上げることができる。CGの問題のマッピング法については今回行なった行方向の分割方式の他にもブロック分割する方法があり、FTは1次元FFTのアルゴリズムと通信のアルゴリズムにおいて改良の余地がある。

6 おわりに

本研究では、超並列計算機CP-PACSのプロトタイプであるPILOT-1の性能評価を、NAS並列ベンチマークを用いて行なった。4つのカーネル問題に対してPILOT-

1に適した並列化を考え、その上で並列動作するプログラムを作成し実測評価を行なった。プログラムはFortran-77言語で記述し、通信はPILOT-1の高速通信機能であるリモートDMA転送を使用して行なった。

EPではほぼ完全な処理の並列化ができ、線形な台数効果と4つのカーネル中最大の性能を得ることができた。MGでは並列度と並列実行可能部分とのトレードオフが見られた。CGでは問題の粒度が台数効果に大きく影響することが確認された。FTでは通信においてHXBの有効性を活かすことができた。

今回の評価ではPILOT-1は相互結合網にHXBを用いていることにより、各種の問題の並列化に対してネットワークのトポロジを意識せずにデータのマッピングが簡単に行え、また多くの転送においてデータ転送を無衝突で行なうことができた。しかし主記憶容量の不足のために実行不可能な問題がいくつか存在した。特にSIOUの主記憶の容量不足は深刻で、多くの問題において16台のノードを全て使えない状況を引き起こした。実機のCP-PACSでは、IOU、SIOUはPUとは独立に設けられるため、この問題は回避されるであろう。

今回は数台程度のノードでの並列化で評価を行ない、ある程度の台数効果が得られたが、更にノード数を増やした時、同様の並列化の方法では台数効果が得られない可能性がある。更に高い性能を発揮させるためと、多くのノードでも台数効果を得ることを可能にするために、データのマッピングや転送方法の改善を考慮することが今後の課題として挙げられる。

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、本研究の一部は創成的基礎研究費(07NP0401)及び文部省科学研究費(奨励(A)07780222)の補助によるものである。

参考文献

- [1] 中澤喜三郎 他：CP-PACSのアーキテクチャの概要、情報研報ARC-108-9, October 1994
- [2] 田中輝男 他：識別子を用いたデータ転送方式を基本とするMIMD型並列計算機アーキテクチャ、並列処理シンポジウムJSPP'89, 1989
- [3] 朴泰祐 他：ハイパクロスバ・ネットワークの性能評価、信学技報CPSPY93-40, pp.41-48, November 1993
- [4] D. Bailey et. al.: THE NAS PARALLEL BENCHMARKS, RNR Technical Report RNR-94-007, March 1994
- [5] K. Itakura et. al.: Preliminary Evaluation of NAS Parallel Benchmarks on CP-PACS, to be appeared in Proc. of PERMEAN'95, August 1995