

## 高精度大規模並列プログラムシミュレーション環境による NPB の挙動解析

久保田 和人<sup>†</sup> 板倉 憲一<sup>‡</sup>  
佐藤 三久<sup>†</sup> 朴 泰祐<sup>‡</sup>

{kazuto, msato}@trc.rwcp.or.jp

itakura@arch.is.tsukuba.ac.jp, taisuke@is.tsukuba.ac.jp

新情報処理開発機構 つくば研究センタ<sup>†</sup>

筑波大学 電子・情報工学系<sup>‡</sup>

### 概要

千台規模の並列計算機上での NAS Parallel Benchmark (NPB) Class B 問題の挙動をシミュレーションした。用いたシミュレーションシステムは、インストルメントツール EXCIT とネットワークシミュレータ生成システム INSPIRE を用いて構築したものであり、並列プログラムの挙動を大規模なプロセッサ数まで、実用的な時間で精度良く解析できる。NPB の 8 つ中の 7 つのベンチマークに関して、プロセッサ台数を 1024 台まで変化させ、キャッシュサイズやネットワークのバンド幅、レイテンシ等の様々なハードウェアパラメータを変化させた場合の挙動を解析した。シミュレーション結果より、全ての問題に関して、キャッシュサイズは実行時間にさほど影響を与えないことがわかった。ネットワーク性能に関しては、EP を除く全ての問題について、通信オーバーヘッドが大きく影響した。バンド幅は、ネットワーク全体の高いスループットが必要な問題 (CG) で影響が表れた。

## Performance Analysis of NAS Parallel Benchmarks on accurate Large-scale Parallel Program Simulation Environment

KAZUTO KUBOTA<sup>†</sup>, KEN'ICHI ITAKURA<sup>‡</sup>, MITSUHI SA TO<sup>†</sup>  
and TAISUKE BOKU<sup>‡</sup>

Tsukuba Research Center, Real World Computing Partnership<sup>†</sup>  
Institute of Information Sciences and Electronics, University of Tsukuba<sup>‡</sup>

### Abstract

We describe the result of a Large-scale data parallel programs (NAS Parallel Benchmark Class B). In our simulation system, message passing trace data of a parallel program is collected by the instrumentation tool EXCIT and it is analyzed by a network simulator generated by INSPIRE. The behavior of parallel programs on thousands of processors can be simulated within a practical simulation time. The influences of various conditions such as cache size and network bandwidth are measured. Experimental results show that a small cache size is enough to effectively execute almost all of the benchmarks. Network start up time affects the total execution time of all benchmarks except EP. However, the influence of network bandwidth is observed only on the CG benchmarks which needs high network throughput.

### 1. ま え が き

近年、CP-PACS<sup>1)</sup>や ASCI-Red<sup>2)</sup>といった数千台規模のプロセッサを持つ高並列計算機 (MPP) が実用化されるようになった。また、PC やワークステーションをイーサネットやギガビット LAN で接続したクラスタシステム<sup>3)4)</sup>も数多く構築されるようになり、並列システムは身近なものとなってきている。

これらの並列システムでは、標準品の CPU が用いられることが多いため、MPP とクラスタ計算機のノードプロセッサの性能差は殆んど無くなってきている。一方、相互結合網に関しては差異が見られ、これが並列システム全体の性

能およびコストを決定している。アプリケーションユーザの視点に立つと、高性能な相互結合網が常に必要というわけではなく、問題によっては安価なもので十分な場合がある。従って、アプリケーションプログラムの特徴を見極め、適切な並列システムを選択することが有効である。本稿では、NAS Parallel Benchmark の CLASS B 問題の挙動をシミュレーションし、並列計算機のハードウェアパラメータを変化させた際の影響について報告する。

Numerical Aerospace Simulation (NAS) の Web サイト<sup>5)</sup>には、各種並列計算機上での NPB の実行時間、および並列化効率に記載されている。これらの結果から、特定の問題を特定の並列計算機で実行する際の適切なプロセッサ台数を、

ある程度決定できる。しかしながら、効率が出ていない場合について、直ちにネットワーク性能の不足と結論づけることはできない。なぜなら、性能低下はネットワーク性能からだけでなく、プログラムの本来の並列性、すなわち、ロードアンバランス、通信の待ち合わせ、逐次計算部分によっても引き起こされるからである。しかしながら、実機を用いて、キャッシュサイズやネットワーク性能を変化させた場合の効率の変化を求めることは難しい。

個々の並列プログラムの並列システムへの適応性、すなわち、ハードウェア性能を変化させた場合の並列プログラムの実行性能の変化を定量的に理解するために、我々はインスツルメントツールとネットワークシミュレータからなるシミュレーションシステムを構築した<sup>7)</sup>。シミュレーション以外の方法として、プログラムを解析的に評価する手法<sup>8)9)</sup>がある。しかしながら、解析だけではプログラムの挙動を詳細に把握することは困難である。なぜなら、プログラムのインプリメンテーションやコンパイラの最適化、ネットワーク上のメッセージの衝突といった実行時の動的要素が考慮されないからである。

我々のシミュレーションシステムでは、トレースデータを生成した後ネットワークシミュレーションを行い精度の良い解析を行う。トレースファイルの作成にインスツルメントツール EXCIT<sup>10)</sup>を用いる。INSPIRE<sup>11)</sup>はネットワークシミュレータ生成システムであり、ネットワーク記述からネットワークシミュレータを生成する。本手法は、以下のような特徴を持つ。

- 大規模なワーキングセット、千台規模の PU(Processing Unit) 台数を持つシステムを予測可能。
- ユーザの持つ実プログラムを評価する。
- 通信と計算のオーバーラップを正確に評価できる。

本手法を NAS Parallel Benchmark(NPB)<sup>6)</sup> ver 2.3 の class B 問題に適用し、PU 台数を千台規模まで変化させた場合について評価を行った。プロセッサ性能に関してはキャッシュサイズを、ネットワーク性能に関しては、トポロジ、バンド幅、通信オーバーヘッドを変化させた。

本稿は以下の構成をとる。第 2 節では、シミュレーション手法の概要について述べる。第 3 節では、シミュレーション結果について述べる。第 4 節では、考察を行う。第 5 節では、まとめと今後の課題について述べる。

## 2. 大規模並列プログラムシミュレーション手法

対象とするプログラムを、通信データの内容によって処理のフローが変化しないデータ並列プログラムに限定する。図 1 は、シミュレーションの流れを示している。

- (1) オリジナルの並列プログラム (prog0-prog3) は EXCIT によってインスツルメントが施され、(prog'0-prog'3) が生成される。
- (2) prog'0-prog'3 は独立に実行され、トレースファイルが生成される。
- (3) トレースファイルを用いてネットワークシミュレーションが行われ、シミュレーション結果が生成される。

EXCIT は、アセンブラレベルのソースコードにインスツルメントを施し、トレースファイルを生成するツールである。EXCIT は、通信関数をダミー関数に置き換える。通信内容にプログラムのフローが依存しないという前提により、ダミー関数に置き換えた後もプログラムは正しいフローで実行され、計算された実行時間は並列プログラムとして実

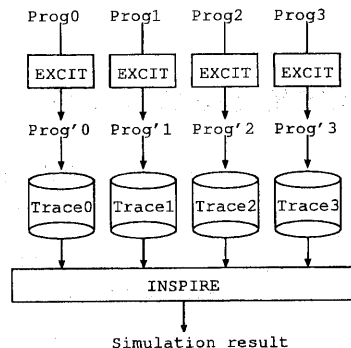


図 1 大規模データ並列プログラムシミュレーションツール

際に行われた場合と、ほぼ同様になる。ダミー関数では、置き換え前の通信関数の情報 (通信相手、サイズ、タグ等) をトレースファイルに格納する。この他に、トレースファイルには通信命令間の処理サイクル数が記録される。

EXCIT では、通信命令間の実行時間を計算するため、ソースコードが基本ブロックに分割される。基本ブロックの先頭にはカウンタが埋め込まれ、実行時に実行回数がカウントされる。load と store 命令の直前では、キャッシュシミュレータと呼ばれる。基本ブロックとキャッシュシミュレーションの情報から、通信命令間の実行時間が計算される。

INSPIRE は、Network Description FILE(NDF) から、クロックレベルのネットワークシミュレータを生成する。NDF は、ネットワーク資源や接続、ルーティングアルゴリズムを記述する。2D-mesh やハイパーキューブ等のネットワーク記述がライブラリとして用意されている。通信バンド幅やオーバーヘッドはパラメータして変化させることができる。生成されたシミュレータにはトレースファイルが入力され、実行サイクル数が計算される。

## 3. NBP 2.3 のシミュレーション結果

### 3.1 前提条件

NPB2.3 の class B 問題を、プロセッサ台数を 1024 台 (LU は 256 台) まで変化させた場合についてシミュレーションした。扱った問題は、EP、MG、CG、FT、LU、SP、BT である。IS は、通信の内容 (ソーティングされるデータの数および中身) に計算内容が依存するので今回は対象外とした。対象とする並列計算機のハードウェアパラメータは以下のように変化した。

- (1) キャッシュサイズ: L1-cache を 8、16、64Kbyte と変化した。表 1 にシミュレーションで用いた CPU のモデルを示す。
- (2) ネットワークトポロジ: 完全結合 (complete), Hyper Crossbar Network(HXB), 2D-mesh および Ring。ルーティングアルゴリズムは、worm-hole routing を用いた。
- (3) ネットワークインタフェースのオーバーヘッド ( $\mu_0$ ): 仮定するネットワークでは、メッセージ送信にあたり定数のオーバーヘッドを要するものとした。 $\mu_0$  を 0、10、500 $\mu$ sec と変化した。
- (4) ネットワークバンド幅 (bw): bw を 10、100、1000 MB/sec、無限大と変化した。

表1 CPUモデル

CPU	2 int pipe, 1 fp pipe
clock	100 MHz
L1-cache(変化)	8192/16384/65536 Byte
L1 miss penalty	5 cycle
L2-cache	1 MByte
L2 miss penalty	20 cycle

表2 EP:計算時間(sec)とキャッシュサイズの関係

cache	16	32	64	128	256	512	1024
8K	90924	45466	22729	11364	5682	2840	1420
16K	90904	45454	22724	11361	5680	2840	1420
64K	90890	45444	22720	11359	5679	2839	1419

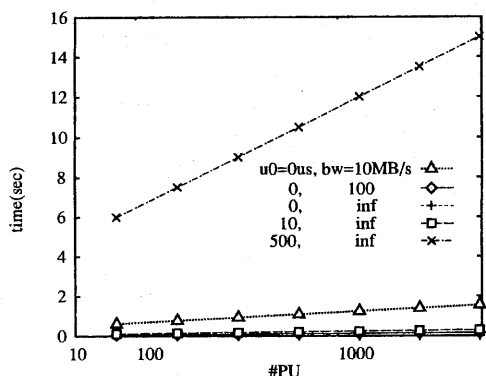


図2 EP:ネットワーク性能と通信時間の関係

### 3.2 実験結果

各々のベンチマークについて、キャッシュサイズを変化させた場合の計算時間を表に示す。ネットワーク性能に関してはパラメータが多岐にわたるため、ここではトポロジーをHXBに、キャッシュサイズを16Kに固定し、bwおよび、 $\mu_0$ を変化させた場合の通信時間をグラフに示した。bwを変化させる時は $\mu_0$ を0に、 $\mu_0$ を変化させる時はbwを無限大とした。これらのパラメータの変更で、通信時間は大きく変化している。HXBは、バンド幅が大きくホップ数が小さいネットワークであるため、他のRingや2D-meshを用いた場合は、bwと $\mu_0$ の影響は更に顕著になる。なお、図および表で抜けている部分は、プラットフォームの都合上ネットワークシミュレーションが行えなかった部分である。

#### 3.2.1 EP

EPは典型的なモンテカルロシミュレーション問題である。コードは、ほぼ完全な並列化が為されている。EPは、探索空間をブロック分割することで並列化を行っている。計算途中には通信は無く、最後に整数10個の配列と2個の実数のリダクションを行っている。

表2に、キャッシュサイズを変化させた場合の計算時間を示す。図2に、ネットワークの性能を変化させた場合の通信時間を示す。 $\mu_0$ を大きくした場合、bwを小さくした場合ともに通信時間は増大しているが、計算時間と比べると無視できる大きさである。

#### 3.2.2 MG

MGは、V-cycle マルチグリッド法を用いてポアソン方程式を解くソルバである。問題領域は  $256 \times 256 \times 256$  のグ

表3 MG:計算時間(sec)とキャッシュサイズの関係

cache	16	32	64	128	256	512	1024
8K	67972	31964	16368	8034	3525	1925	1049
16K	63401	32246	15697	7812	4122	1914	1042
64K	61612	28788	15541	7758	3380	1855	1010

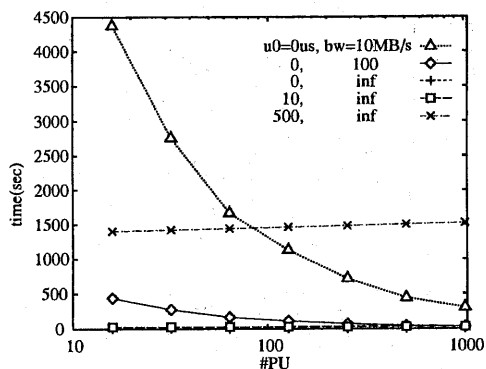


図3 MG:ネットワーク性能と通信時間の関係

表4 CG:計算時間(sec)とキャッシュサイズの関係

cache	16	32	64	128	256	512	1024
8K	414405	209352	115107	55580	35535	16509	11569
16K	410370	205804	113398	54114	34768	15938	11275
64K	392492	190885	105973	48793	31879	14136	9945

リッドである。並列化に際しては、各PUの担当領域なるべく立方体に近くなるようにグリッドをブロック分割する。アルゴリズム中のVサイクルでは、グリッドを全ての軸方向について半分にしながら計算を行うが、分割不可能になった時点で代表PUによる並列化が行われる。すなわち、最も粗いフェーズでは  $2 \times 2 \times 2 = 8$ PUによる並列化が行われる。通信は、並列処理をしているPU間での隣接通信となる。

表3に、キャッシュサイズを変化させた場合の計算時間を示す。図3に、ネットワークの性能を変化させた場合の通信時間を示す。bwを変化させた場合のグラフは右下がりとなる。一方 $\mu_0$ を変化させたグラフは平坦になる。MGでは、PU台数によらずメッセージの数は、ほぼ一定である。したがって、 $\mu_0$ が大きい場合グラフ全体が上方向にシフトすることになる。一方、メッセージサイズはPU数が小さい場合に大きいので、バンド幅が小さい場合、通信時間が大きくなる。計算時間と比較すると、bw = 10, 100 MB/secともに問題ない。一方、 $\mu_0$ に関しては、10  $\mu\text{sec}$ なら問題ないが、500  $\mu\text{sec}$ では並列化効率が悪化する。

#### 3.2.3 CG

CGは、Conjugate Gradient法を用いて大規模疎行列の最小固有値を求める問題である。問題サイズは、75000である。並列化は、疎行列Aを2次元に分割することで行われる。PU数が平方数でない場合は列を2倍のPU数で分割する。

表4に、キャッシュサイズを変化させた場合の計算時間を示す。図4に、ネットワークの性能を変化させた場合の通信時間を示す(bw = 10 MB/secの場合は表5に示す)。グラフが上下に振動しているのは、PUの分割方法による影響

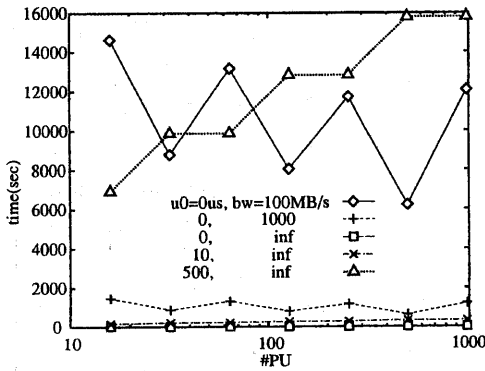


図4 CG:ネットワーク性能と通信時間の関係

表5 CG:通信時間(sec)( $\mu_0 = 0$ ,  $bw = 10$  MB/sec)

PU=16	32	64	128	256	512	1024
146260	87763	131639	80462	117015	62185	-

表6 FT:計算時間(sec)とキャッシュサイズの関係

cache	16	32	64	128	256	512	1024
8K	380600	193067	95262	47727	24194	20209	8895
16K	378340	191846	94698	47445	24060	19865	8799
64K	372339	188908	93197	46695	23824	16149	7855

表7 FT:通信時間(sec)( $\mu_0 = 0$ ,  $bw = 10$  MB/sec)

PU=16	32	64	128	256	512	1024
-	34131.5	17341.6	8740.77	4390.04	-	-

と考えられる。bwを変化させたグラフは、PUが増えても通信時間が減少しない。これは、CGではPU台数が増えるとメッセージサイズは減るが、逆に通信回数が増えるためである。 $\mu_0$ を変化させたグラフはPUの増加に伴って通信回数が増えるため右上がりとなる。bw = 10 MB/secだと16PUの場合で通信時間が計算時間の35%になる。bw = 100 MB/secでも通信時間が見え、1 GB/secでようやく見えない程度となる。 $\mu_0$ に関しては、10 $\mu$ secならば見えないが500 $\mu$ secだと1024台の場合、計算時間を超える大きさとなる。

### 3.2.4 FT

FTは、Partial Differential Equation(PDE)をFFTおよび逆FFTを用いて解く問題である。問題サイズは、512 $\times$ 256 $\times$ 256である。並列化は、Z次元をブロック分割することで行う。PUが余る場合は、Y次元もブロック分割を行う。すなわち、PU空間は1次元か2次元となる。1次元のPU空間の場合は、X次元のFFT、Y次元のFFT、転置、Z次元のFFTという処理となる。2次元のPU空間の場合は、転置、X次元のFFT、Y次元のFFT、転置、Z次元のFFTという処理となる。

表6に、キャッシュサイズを変化させた場合の計算時間を示す。図5に、ネットワークの性能を変化させた場合の通信時間を示す(bw = 10 MB/secの場合は表7に示す)。 $\mu_0$ のグラフが、PU=512および1024で平坦になっているのは、512台以降、PU分割方式が変わるためである。256台までは通信時間殆んど見えない。bwのグラフは右下がりであり、10 MB/secでも計算時間に対して少し見える程度である。100

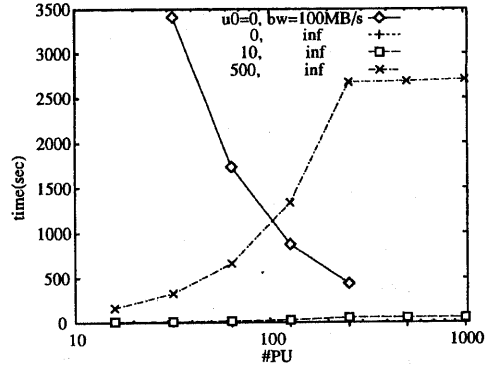


図5 FT:ネットワーク性能と通信時間の関係

表8 LU:計算時間(sec)とキャッシュサイズの関係

cache	16	32	64	128	256
8K	1197317	601296	292704	155492	81898
16K	1187023	592827	288546	154278	81191
64K	1174804	586174	284162	150027	77394

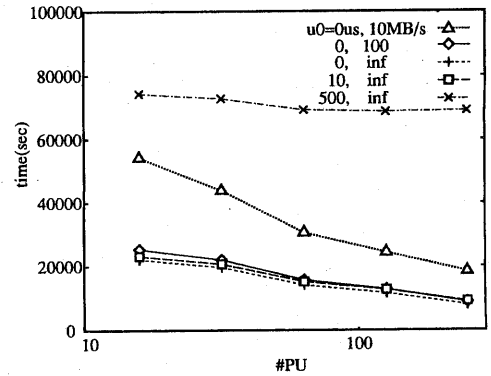


図6 LU:ネットワーク性能と通信時間の関係

MB/secだと、殆んど影響が無くなる。

### 3.2.5 LU

LUは、SSOR(Symmetric Successive Over-Relaxation)法を利用したNavier-Stokes方程式のソルバである。Class Bでは、問題サイズは102 $\times$ 102 $\times$ 102である。

表8に、キャッシュサイズを変化させた場合の計算時間を示す。図6に、ネットワークの性能を変化させた場合の通信時間を示す。bwのグラフが右下がり、 $\mu_0$ が平坦というのは他のベンチマークの傾向と同様である。bw = 10 MB/secでも、 $\mu_0 = 500$   $\mu$ secの場合を常に下回っている。これは、PU台数が小さい場合もメッセージサイズがさほど大きくないことによる。bw = 100 MB/secのグラフは、無限大の場合とほぼ同程度の性能を示している。

### 3.2.6 SP、BT

SPとBTは、multi-partitionアルゴリズムを利用したNavier-Stokes方程式のソルバである。

表9に、キャッシュサイズを変化させた場合のSPの計算時間を示す。図7に、ネットワークの性能を変化させた場合

表9 SP:計算時間(sec)とキャッシュサイズの関係

cache	16	64	256	1024
8K	1213966	286587	76334	23518
16K	1164691	283912	75137	22306
64K	1144390	279403	71834	20933

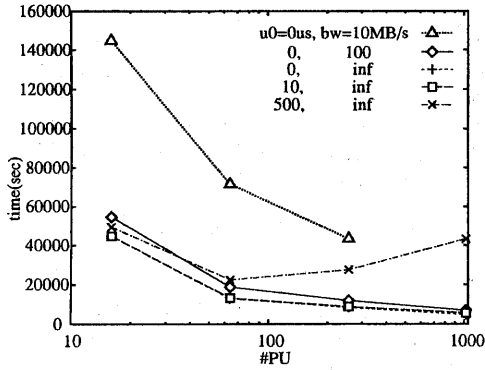


図7 SP:ネットワーク性能と通信時間の関係

表10 BT:計算時間(sec)とキャッシュサイズの関係

cache	16	64	256	1024
8K	1570817	382718	97954	26989
16K	1544499	381301	97191	26434
64K	1540258	378898	95545	26137

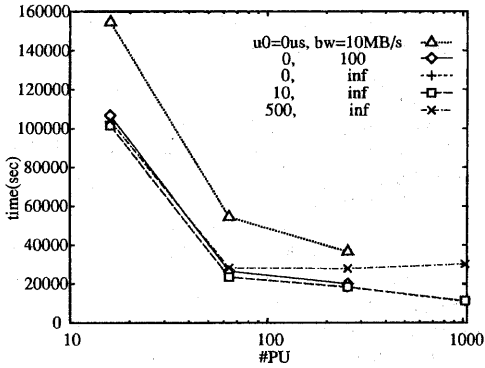


図8 BT:ネットワーク性能と通信時間の関係

のSPの通信時間を示す。表10に、キャッシュサイズを変化させた場合のBTの計算時間を示す。図8に、ネットワークの性能を変化させた場合のBTの通信時間を示す。bwのグラフは右下がりとなる。bw = 100MB/secならば、通信時間は $\mu_0 = 0$ 、bw = 無限大というほぼ理想的なネットワークと同等となる。しかしながら、この場合でもある程度の通信時間がかかっている。これは、両方のプログラムとも、プログラム自体に潜在的な逐次性が含まれているとためだと考えられる。

#### 4. 考察

##### 4.1 プログラムが潜在的に持つ逐次性

オーバーヘッド無し、バンド幅無限大という理想的なネッ

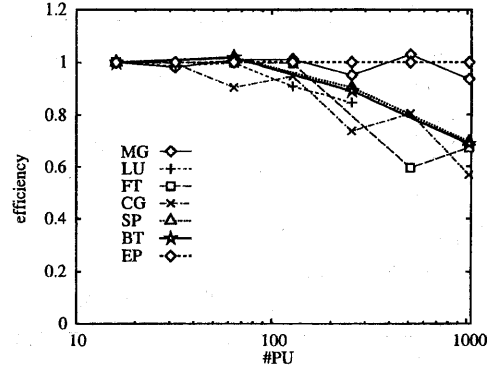


図9 理想的なネットワークを用いた場合の並列化効率

トワークの場合において、いくつかのベンチマークはスケールアップしなかった。図9は、理想的なネットワーク時ににおける並列化効率をグラフにしたものである。EPとMGを除いて1024台の場合の並列化効率は0.7程度に落ちている。これは、負荷分散のアンバランス、プログラムの構造的なメッセージの待ち合わせ、並列化できない逐次部分の処理といった、個々の並列プログラムに元々含まれている逐次性によるものだと考えられる。したがって、このような場合ネットワーク性能を向上させても効率の向上は頭打ちとなる。

##### 4.2 キャッシュの影響

キャッシュサイズは、CGを除くほとんど全てのベンチマークで影響が小さかった。プロセッサ16台でキャッシュサイズが8Kbyteの場合、L1キャッシュのミス率は、MG、SPでそれぞれ、8.9%、12.7%であった。プロセッサ台数が増えた場合、プロセッサあたりのワーキングセットが小さくなるのでキャッシュサイズの影響は小さくなる。FT、BT、EPに関しては、キャッシュの影響は殆んど認められなかった。CGに関しては、16台の場合キャッシュミス率は25%であった。L1-cacheのラインサイズは32byteを仮定しているので8バイトのdoubleの連続アクセスが全て外れているという状況だと想定できる。しかしながら、この場合でも全体の計算時間に対するキャッシュミスの影響は僅かであった。

##### 4.3 ネットワークの影響

ネットワークパラメータの影響について考察する場合は、常に計算時間と通信時間の比を考慮する必要がある。殆んどどのプログラムでもPU台数が増えると計算時間は反比例して減少する。したがって、PU台数が少ない場合は、ある程度の通信時間を要しても全体の計算時間には反映されない。逆にPU台数が増加した場合は、通信時間が減少していかないと全体の計算時間に通信時間が見えてきてしまい、台数効果が出ないことになる。

##### ● バンド幅(bw)の影響

バンド幅の影響は各々のプロセッサの通信量に依存する。殆んどどのベンチマークでも、プロセッサ台数が増加すると各々のプロセッサが通信するデータの量は小さくなる。したがって、トータルのスループットが十分確保できるトポロジーを選択すれば、バンド幅の影響は小さいと言える。一方CGのような問題では、一台あたりの通信量はプロセッサ台数が増えても減少しないので、バンド幅の影響が出る。

##### ● オーバーヘッド( $\mu_0$ )の影響

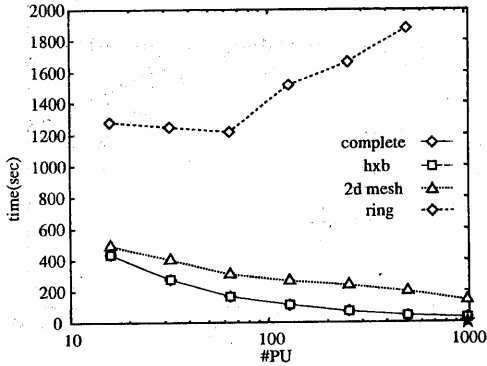


図 10 MG: トポロジと通信時間の関係

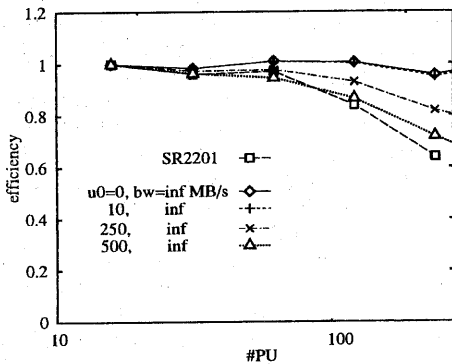


図 11 MG: SR2201 との効率の比較

オーバーヘッドは、プロセッサ台数が増大すると影響が大きくなる。これは、プロセッサ台数が増加した場合、各プロセッサの計算時間が減少するのに対して、通信回数は一定か、あるいは増大するからである。

#### ● トポロジの影響

図 10 に、MG について、トポロジを完全結合、HXB、2D-mesh、Ring と変化した場合の通信時間を示す ( $\mu_0=0$ ,  $bw=100\text{MB/sec}$ )。MG は、三次元の隣接通信を行うので、プロセッサ台数が増えるにしたがって大きなネットワークスループットが必要となる。完全結合に対して、HXB はほぼ同程度の性能を示している。2D-mesh も健闘しているが、スループットの低い Ring はプロセッサ台数が増えるにしたがって、通信時間が大幅に増えているのがわかる。

#### 4.4 実機との比較

MG に関して、SR2201 との実行効率の比較を行った (図 11)。SR2201 は CPU が 150MHz、MPI レベルの  $\mu_0$  と  $bw$  の実測値がそれぞれ  $27.3\mu\text{sec}$ 、 $214.8\text{MB/sec}$  である。今回シミュレーションで用いたパラメータで正規化すると、 $\mu_0=41.0\mu\text{sec}$ 、 $bw=143.2\text{MB/sec}$  となる。 $bw=100\text{MB/sec}$  とし、 $\mu_0$  を 10、250、 $500\mu\text{sec}$  としたグラフと比較すると、実機の性能は 128 台以降、急激に下がり始めている。プロセッサモデルその他の環境が異なるので直接比較はできないが、この差異の原因を明らかにすることが今後の課題の一つとなる。

## 5. まとめと今後の課題

本稿では、NPB 2.3 class B をプロセッサ台数を 1024 台まで変化した場合についてシミュレーションを行った。通信オーバーヘッドは、ほぼ全ての問題について大きな影響を及ぼした。バンド幅は、CG にのみ大きな影響が現れた。理想的なネットワークを使った場合でも、並列化効率は常に 1 となるとは限らず、いくつかのベンチマークでは、1024 台の時に 0.7 程度まで下がった。これらの実験結果は、アプリケーションユーザが最適なプロセッサ台数やネットワークを選択するのに有益な情報となる。また、アーキテクチャ設計者にも重要な指針を示すことができる。今後は、実機との比較、精度の検証、メモリバンド幅といった他のパラメータを用いたシミュレーション、および、ベンチマークプログラムの分類および特徴づけを行う予定である。

## 謝 辞

日頃、性能評価技術に関し議論して頂く TEA グループのメンバー諸氏に感謝いたします。TEA グループは、研究技術組合新情報処理開発機構・電総研・筑波大学を中心とする性能評価に関する研究グループである。

## 参考文献

- 1) A. Ukawa, "Status of the CP-PACS Project," International Symposium on Lattice Field Theory, 1994.
- 2) T.G. Mattson, D. Scott, and S.R. Wheat, "A TeraFLOP Supercomputer in 1996: the ASCI TFLOPS System," Proceedings of the International Parallel Processing Symposium, 1996.
- 3) <http://now.cs.berkeley.edu/>
- 4) <http://www.rwcp.or.jp/lab/pdslab/clusters/home.html>
- 5) <http://science.nas.nasa.gov/Software/NPB>
- 6) David Bailey, Tim Harris, William Saphir, Rob van der Wijngaart, Alex Woo, and Maurice Yarrow, "The NAS Parallel Benchmarks 2.0," NASA Ames Research Center Report, NAS-05-020, 1995.
- 7) 久保田和人, 板倉憲一, 佐藤三久, 朴泰祐, "大規模データ並列プログラムの性能予測手法と NPB2.3 の性能評価", JSPP'98, pp.71-78, 1998.
- 8) Maurice Yarrow and Rob Van der Wijngaart, "Communication Improvement for the LU NAS Parallel Benchmark: A Model for Efficient Parallel Relaxation Schemes," NAS Technical Report NAS-97-032, 1997.
- 9) Edward Rothberg, Jaswinder Pal Singh, and Anoop Gupta, "Working Sets, Cache Sizes, and Node Granularity Issues for Large-Scale Multiprocessors," Proc. of ISCA'93, pp.14-25, 1993.
- 10) 久保田和人, 板倉憲一, 佐藤三久, 朴泰祐, "インストレーションに基づく性能評価環境の高精度化", 情報研報, ARC123-12, pp.67-72, 1997.
- 11) Taisuke Boku, Tomoaki Harada, Takashi Sone, Hiroshi Nakamura, and Kisaburo Nakazawa, "INSPIRE: A general purpose network simulator generating system for massively parallel processors," Proc. of PERMEAN'95, pp.24-33, 1995.