高並列計算機の性能評価のための挙動予測モデル

古市実裕[†,☆] 永松礼夫[††] 出口光一郎[†]

並列計算機システムを効率的に利用するには，性能挙動全般を知ることが重要であるが，並列計算機の性能は複数の要因の相互作用で複雑な挙動を示すため，その概要を把握することでさえ容易ではない．本稿では，並列度やキャッシュ容量・方式，通信コストなどの性能を決定づける個々の要因ごとの簡単なモデルを組み合わせることで，全体の性能の挙動を予測する手法を提案する．また，実機を用いた実験で，本手法での挙動予測モデルが実際の性能をよく表現し，性能予測が可能であることを確認した．

An Estimation Model for Performance Evaluation of Highly Parallel Computer

SANEHIRO FURUICHI,[†,☆] LEO NAGAMATSU[††] and KOICHIRO DEGUCHI[†]

It is very important for the effective use of highly parallel computer to understand its overall performance. But it depends on many factors, and it shows very complicated behavior. We propose a new method to predict the overall performance using simple models built from a combination of the degree of parallelism, memory caching mechanism and capacity, communication cost and so on. Experimental results of performance estimation on some real highly parallel computeres show the feasibility of this method.

1. はじめに

より高速な計算機環境を実現する手段として，並列計算機は非常に有望である．ところが，構成要素があまりにも複雑で，計算機の性能を詳細に把握するのが困難なため，性能を最大限に生かした利用は難しい．ベンチマークの結果やスペックは，ある程度性能の傾向を知るヒントとはなるが，実際にユーザが直面する利用形態を想定しているわけではなく不十分である．近年，あらゆる利用形態での性能を何とか知ることができないか，あるいは何らかの指標で表せないかという研究が進められている[1),2)]．本論文ではその試みの1つとして，並列度やキャッシュメモリの容量，通信バンド幅などの性能を決定づける要因を特定し，要因ごとの簡単なモデルを組み合わせて全体の複雑な性能を予測する手法を提案する．また，本提案を複数の実機へ適用し，評価を行った例もあわせて紹介する．

† 東京大学工学部
Faculty of Engineering, University of Tokyo
†† 会津大学情報センター
Information Systems and Technology Center, University of Aizu
☆ 現在，日本アイ・ビー・エム株式会社東京基礎研究所
Presently with Tokyo Research Labolatory, IBM Japan

2. 速度向上率による評価とその問題点

並列計算機の性能評価には，従来から速度向上率がよく用いられている．プロセッサ数 N で問題サイズ W の問題を解く所要時間を $T(N,W)$ としたとき，同じ問題を1台のプロセッサで解く場合と比べた速度向上率 $S(N)$ が，

$$S(N)=\frac{T(1,W)}{T(N,W)} \tag{1}$$

と定義され，プロセッサ数が増えたときにどれだけ時間が短縮されるかを示す指標となる．ただし，問題サイズ W は，ここでは必要な演算回数と定義する．また，W は何らかの条件で与えられる値で，パラメータではない．投資したプロセッサ数の分だけ時間短縮が期待され，この値は理想的には N になるはずである．この理想にどれだけ近いかを表す指標として，

$$E(N)=\frac{S(N)}{N} \tag{2}$$

を効率と称して評価指標にすることもある．

従来，この評価は問題サイズをつねに一定にして行われるのが一般的だった（fixed-size speedup と呼ばれる）[3)]．しかしプロセッサ数に見合った問題サイズで評価しないと性能を過小評価する恐れがあるとし

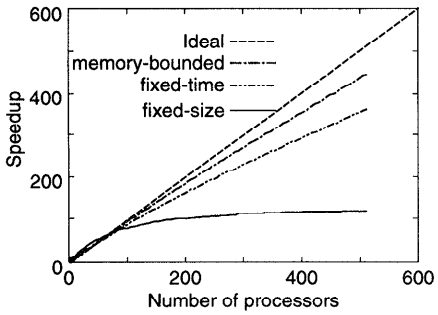


図1 AP1000で行列乗算実行時の速度向上率．問題サイズ固定のままより，システム規模に応じてサイズを増やす方が高い評価値を示す．

Fig. 1 Speedups of matrix multiplying on AP1000. The fixed time speedup and the memory bounded speedup indicate higher values than those of the fixed size speedup.

て，近年，実行時間を一定に保つように問題サイズ W を変化させて評価する方式（fixed-time speedup）[4]や1PE（プロセッサ要素）あたりのメモリ使用量を一定に保つように W を変化させる方式（memory-bounded speedup）[5]などが提案されている．

図1に，並列計算機AP1000で行列乗算を実行したときの上記の方式での各速度向上率を示す．基準は1PEで 1024×1024 行列の乗算を実行したときの性能である．N の増加によりどれだけ時間短縮されるかを表す．

ところが，速度向上率による性能評価には，以下のような問題点がある．

- $N = 1$ での性能がメモリ容量不足などのために測定不可能となる場合が多い☆．
- 特定アルゴリズムの1PEでの性能を基準にするため，並列計算に向いた他のアルゴリズムとの比較が困難．
- 特定の (N, W) でしか評価せず，任意の利用形態での性能は不明．

これらを踏まえたうえで，以下の章では，適切な評価指標のパラメータや評価基準値について見直しを行い，複雑な性能挙動をいかにとらえるかを考察する．

3. 曲面による性能評価

速度向上率の大きな問題点は，任意の利用形態での性能を知るには不十分な点である．ここでは，評価指標のパラメータに問題サイズ W を加えることで，より詳細な評価が可能となり，性能を最大限に生かした利用のための指針が得られることを論じる．

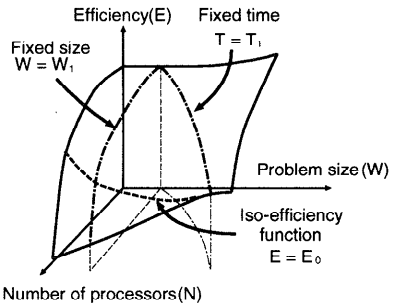


図2 プロセッサ数 N と問題サイズ W の関数として表現された効率．従来は断面の一部しか評価していなかった．また，曲面が単調な場合には，ある特定の等高線形状（iso-efficiency関数）を求めることで全体を表現することが可能である．

Fig. 2 The efficiency surface described as a function of processor number N and problem size W. Conventional methods evaluated only a part of cross section of this surface. If this surface were monotonic, it could be possible to express a contour of the iso-efficiency function as a function of N and W.

3.1 効率と効率曲面の定義

性能評価指標として，効率 $E(N,W)$ を，

$$E(N,W) = \frac{V(N,W)}{V_{\mathrm{ref}}} \qquad (3)$$

$$\left(\begin{array}{l} V(N,W) = \dfrac{W}{NT(N,W)} \\ V_{\mathrm{ref}} = \text{基準速度 (PE ピーク性能)} \end{array} \right)$$

と定義する．式(2)の定義との違いは，パラメータが N と W に増えたことと，評価基準をアルゴリズムに依存しないプロセッサ単体のピーク性能にとった点である．

パラメータが増えたことにより，効率は**図2**のように空間中の曲面として表現される．これを効率曲面と呼ぶことにする．従来の問題サイズや実行時間を一定にした速度向上率や効率は，図中に示すように効率曲面の断面の曲線を評価したにすぎない．我々の目的は，曲面の全体形状を求めることにある．

次節で，関連する研究について簡単に述べた後，曲面形状を求める新たな手法を提案する．

3.2 関連する研究

性能を図2のように曲面で表現し，曲面形状を評価することで挙動をとらえようとするいくつかの研究が進められている．

Kumarらによるiso-efficiency関数[6]は，効率を一定値 E に保つのに必要な問題サイズを N の関数として，

☆ 実際，AP1000の例でも，fixed-time speedupとmemory-bounded speedupでは $N \leq 16$ でしか実測できず，図1は後述の予測モデルにおいてメモリ容量無限大を仮定して計算した．

$$W = f_E(N) \tag{4}$$

と表したもので，図2中の点線で示すように，曲面の等高線の形状を表す関数に相当する．彼らは，たとえば $E = 0.5$ の等高線形状を求め，近似的に曲面全体の形状を推測している．この手法は曲面形状の単調性を前提としており，曲面に局所的な不連続性がある場合などには適用できない．一般に効率曲面は，アーキテクチャやアルゴリズムの様々な要因によって複雑な形状をするので，適用できない場合が多い．

同様な提案として，Xian-He Sun らの iso-speed[7] がある．これは1PEあたりの平均計算速度

$$V(N,W) = \frac{W}{NT(N,W)} \tag{5}$$

を一定に保つ条件を求めるもので，問題サイズ固定で評価するという慣習からは脱しているが，基本的に iso-efficiency と同様の問題を抱える．

また，関口らによるスケーラビリティ関数[2]は，性能全体の振舞いを知るために，性能評価指標の微分をとったもので，利用形態に応じて性能がどちらの方向へ進みつつあるのかを容易に読みとろうというものである．評価指標が全範囲で微分可能であると仮定しており，数カ所の実測結果と組み合わせて補間することで，任意の利用形態における未知の性能を予測することが可能であるとしている．

4. 効率曲面の形状予測

前章で紹介した手法は，利用形態の全範囲を視野に入れたうえで，トップダウン的に性能の挙動をとらえようとする試みである．しかし，グローバルな視点からの挙動はとらえられても，局所的な変動などには十分対応できない場合もある．また，効率曲面全体の形状を知るために，(N,W) の全組合せを実測するのも現実的ではない．

ここでは，**図3**に示すように，性能を決定づける主たる要因を特定し，個々の要因ごとの簡単な性能モデルを組み合わせてボトムアップ的に効率曲面の全体像を予測する新しい手法を提案する．また，次章では，複数の並列計算機上でのプログラムに本手法を適用した例を紹介し，本手法による性能予測が可能なことを確認する．

4.1 性能予測モデルの提案

性能を決定づける要因が並列度のみであれば，性能挙動も単調で，任意の利用形態での性能を簡単に予測できる．しかし，現実にはキャッシュメモリの容量や方式，ネットワークの特性，アルゴリズムの違いなどによって，効率の局所的な山や谷が生じ，その曲面形状は複雑になる．

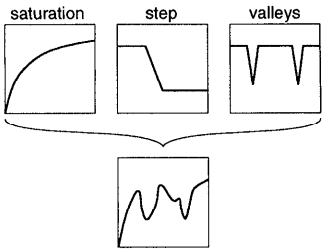


図3 並列度の限界による飽和，キャッシュ容量の制限による段差，キャッシュメモリや主メモリの方式による谷などの，個々の要因から全体の性能を予測する．

Fig. 3 The overall performance estimation by means of combination of each factors such as saturation of parallelism, memory access gap by cache capacity, and valleys by method of cache and main memory accesses.

ここでは，個々の要因ごとに簡単なモデルをたて，各々を組み合わせることで全体の複雑な形状を予測する手法を提案する．本手法では，予測モデルに必要なパラメータを決定できる範囲の小規模なシステムでのみ実測を行い，大規模システムでは実測することなく全体の性能を予測する．

4.2 具体的な予測モデル

性能を決定づける主たる要因を，1PEあたりの問題サイズ，プロセッサ単体の性能，キャッシュメモリの性能と容量，プロセッサ間の通信バンド幅であると特定する．並列実行時間 $T(N,W)$ を，計算部分 T_c とオーバヘッド部分 T_o に分け，

$$T(N,W) = T_c(N,W) + T_o(N,W) \tag{6}$$

とする．プログラムコードなどから，1PEあたりの問題サイズ w を求めることで，

$$T_c = aw \tag{7}$$

と見積もれる．ただし，a は1演算に要する計算時間である．ここで，キャッシュメモリのヒット・ミスの頻度とこの演算に要する時間 a との関係を**図4**のようにモデル化する．a は問題サイズ w の値に応じてキャッシュメモリの容量・構成の関係から段階的に変化し，効率曲面にも段差が生じる．ここで，a の値の挙動を表す図中のパラメータ a_{miss}，a_{hit}，W_{h}，W_{m} などは，比較的少数のPEによるシステムにおいて実測可能であることが重要である．

一方，オーバヘッド T_o に関しては，ここでは主要な要因は通信コストであると仮定する．すると，メッセージサイズ L，通信回数 m，通信バンド幅 b として，この実行時間は

$$T_o = \frac{mL}{b} \tag{8}$$

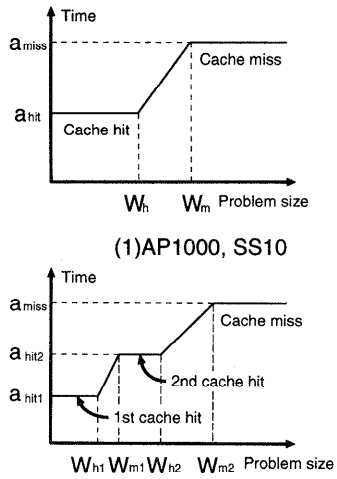


図4 キャッシュメモリのモデルの例．1PEあたりの問題サイズ w の大きさに応じて，キャッシュメモリに収まるときと収まらないときとで計算時間が段階的に変化する．AP1000とSPARCstation10は(1)，Cenju-3は(2)のモデルを用いた．

Fig. 4 Examples of cache model. Computation time of one operation changes whether cache is hit or missed depending on the problem size for 1 PE.

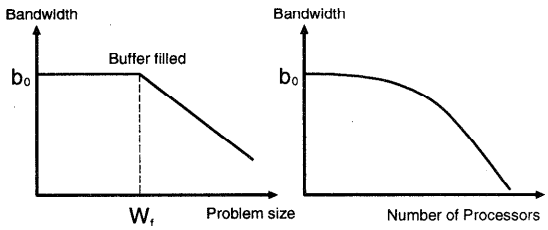


図5 バンド幅のモデルの例．結合網の構造やアルゴリズムに応じて，W に依存する場合や，N に依存する場合，あるいは両方に依存する場合もある．これらはアーキテクチャやアルゴリズムに応じて使い分ける．

Fig. 5 Examples of bandwidth model. It depends on processor number N, problem size W, network architecture, and computation algorithm.

とモデル化できる．L, m, b はアルゴリズムやアーキテクチャに応じて，N と W に依存する．キャッシュメモリの場合と同様，バンド幅 b もプロセッサ数やネットワークの形態，同期のタイミングなどに応じて変化し，**図5** に示すような様々なモデルをたてられる．ここでも a の場合と同様に，これらのパラメータは比較的小規模なシステムで実測可能である．

また，式(6)のように，いつでも計算部分とオーバヘッド部分に分割できる場合ばかりとは限らず，たとえばよく行われているように，計算と通信をオーバラップさせる場合などは，単純な和ではなく両者のMaxをとるなどの調整も必要である．また，プロセッサ間の負荷が均一でない場合などにも同様の処置が必要である．

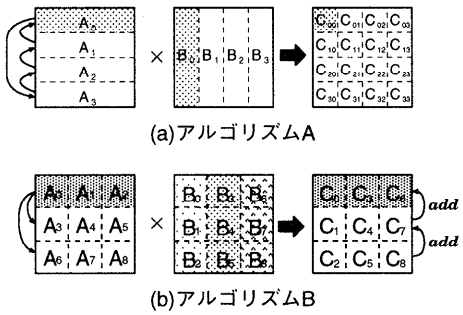


図6 実機での性能予測に用いた行列乗算アルゴリズム．演算回数は同じだが，通信回数が異なる．

Fig. 6 Matrix multiplying algorithms used in the experiment on real parallel computers. The total number of operations are same for both of them, but their communication times are different.

5. 実機を用いた性能予測

本章では，提案した手法を実際の高並列計算機に適用し，その有効性を評価した例を示す．実験にはAP1000（1～512PE），Cenju-3（1～256PE），SPARCstation10のワークステーション・クラスタ（1～64台）を用いた．並列プログラムには，**図6** に示す $n \times n$ 行列乗算アルゴリズム2種類を用いた．

まず，アルゴリズムAを3種類の計算機上で実行した場合の性能予測の結果を示す．その結果を用いて，アーキテクチャやアルゴリズムに応じた適切な問題サイズやプロセッサ数を選択するヒントが得られることも示す．次に，同じ計算機（AP1000）で異なるアルゴリズムBを実行した場合の性能予測と，アルゴリズムAとの性能比較の例を示す．上記のパラメータの値が，アルゴリズムにも応じて変化することを示す．

5.1 行列乗算（アルゴリズムA）

(a) 予測モデル

$n \times n$ 行列乗算をプロセッサ数 N のシステムで実行した場合の予測モデルを，

$$T_c = a \times \frac{n^3}{N^2} \times N = an^3/N$$

$$T_o = \frac{1}{b} \times \frac{n^2}{N} \times (N-1)$$

とした．1演算の計算時間 a，バンド幅 b の値は，その前後の準備処理等も全部ひとまとめにして係数に含めているため，厳密な意味での計算時間，バンド幅とは異なる場合もある．それぞれのモデルは計算機により異なり，各モデルに必要なパラメータはプロセッサ

表1 AP1000における予測モデルのパラメータ
Table 1 Parameters in estimation model of AP1000.

CPU		SPARC 25 MHz
基準速度 V_{ref}		2.78 (MFLOPS)
1ステップの計算時間 a	a_{hit}	0.71 (μs)
	a_{miss}	1.20 (μs)
キャッシュサイズ		256 (KB)
通信バンド幅 b		2.0 (B/μs)

数台程度の小規模システムでの実測から決定する．また，ピーク性能に関しても，メモリからロード，倍精度浮動小数点演算（乗算），ストアという過程を繰り返し実行したときの計算速度の実測値の最大値としたので，純粋な演算だけの性能である文献値とは異なる．

ここでは，これらのパラメータを次のような方法で求めた．まず，プログラムコード中の計算処理の前後や通信処理の前後に時間計測用のシステムコールを適宜挿入し，このプログラムを小規模システムで実行して T_c や T_o の大まかな値のオーダーとその比を求め，実行時間 $T = T_c + T_o$ の簡単なモデルを作成した．また，同時に計測に必要な (N, W) の範囲も決めた．続いて，時間計測用のシステムコールを抜いた本来のプログラムで全実行時間 T を測定し，パラメータ a，b を含む，実行時間 T のモデルにあてはめ，最もフィットする係数をモデルのパラメータとして決定した．

(b) AP1000

予測モデルのパラメータはプロセッサ数1～8の小規模システムでの実測結果より決定し，512台までの性能を予測した．測定したモデルパラメータを**表1**に示す．各PEは1次元トーラス上で隣接PEとのみ通信するため，メッセージの衝突はなく，通信バンド幅は N や W に依存しない一定値となった．

モデルから推測した効率曲面を**図7**に示す．問題サイズを増やしていくと，キャッシュ容量の制限のために性能が落ちる様子や，プロセッサを増やしたときの効率の低下度合などが読みとれる．予測モデルにより求めた効率曲面が正しく予測されているかを検証するために，これとは別に，すべての (N, W) の点で効率を実測して求めた効率曲面を**図8**に示す．より詳細に比較するために，**図9**には，各々の曲面の $N = 32$ における断面図を重ねて表示する．十分良い精度で予測できていることが示されている．性能を決める要因が把握できれば，簡単なモデルでも十分実用的なレベルで性能予測が行えることが分かる．

また，この効率曲面を等高線表示したものを**図10**に示す．図中＊印はメモリ容量の限界点であり，実機ではこのラインより上部では実行不可能である．図で

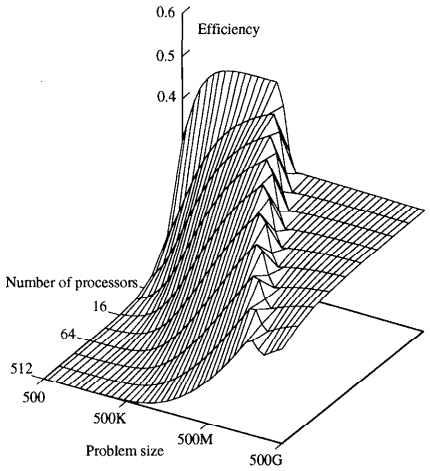


図7 行列乗算プログラムの予測効率曲面（AP1000）．キャッシュの影響や台数効果を読みとれる．
Fig. 7 Estimated efficiency surface for matrix multiplying on AP1000. It shows cache effects and processor number effects.

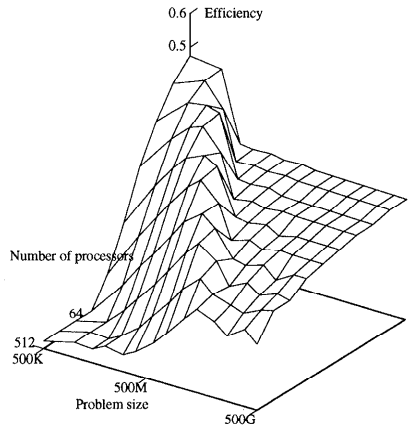


図8 実際に測定した効率曲面（AP1000）．
Fig. 8 Measured efficiency surface for matrix multiplying on AP1000.

示されるように，曲面には尾根が存在しており，尾根に沿って極大性能となる利用形態をとれば，ある台数のPEに対する性能を最大限に引き出せることが分かる．また，P点近傍のように，問題サイズやプロセッサ数がわずかに変化するだけで急激に性能が低下する場合があり，注意を要することも分かる．さらに，たとえば現状がP点でプロセッサ数が10台程度であり，システム規模を拡大しようと考えている場合，同じ問題を解くのにプロセッサ64台程度までなら投資効果もあるが，それ以上増やしても効率が下がり，コストに対して見合った大きな効果が得られにくくなるなどの情報も読みとれる．このように，性能の全般的な挙

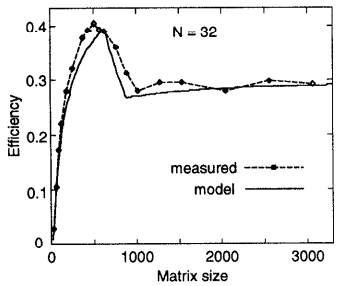


図9 AP1000のモデル予測値と実測値との比較（$N = 32$における断面図）．簡単なモデルでも要因を特定できれば十分実用的なレベルの予測が可能である．

Fig. 9 A comparison of an estimated performance and a measured performance for the cases of $N = 32$. These two are sufficiently identical, which conclude that such a simple model proposed here produces good performance estimations if proper factors are employed to consideration.

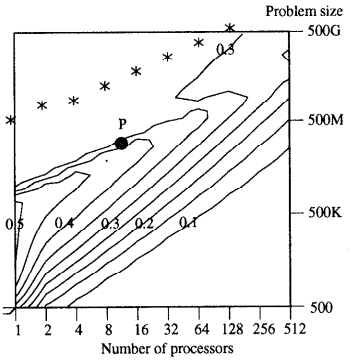


図10 AP1000の効率曲面の等高線表示．適切なプロセッサ数や問題サイズ，スケジューリングのヒントなどの情報を含む．＊印はメモリ容量による計算の限界点を表す．

Fig. 10 Contour map of the efficiency surface of AP1000. It describes sugestions for suitable processor number, problem size, scheduling hints and so on. *s describe a computation limit by real memory capacity.

動をつかめると，システムを利用するうえで様々な有用情報を入手できる．

(C) Cenju-3

同じアルゴリズムをCenju-3で実行した場合のモデルによる性能予測の例を紹介する．プロセッサ1～16台での実測結果を用いて予測モデルのパラメータを決定し，256台までの性能を予測した．このシステムは2次キャッシュも持つため，キャッシュモデルは図4(2)を用いた．実測したパラメータを**表2**に示す．

各プロセッサ間は多段結合網で結合されており，メッセージの転送距離はつねに一定だが，各スイッチング・ユニットでメッセージの衝突が生じ，プロセッサ数の増加とともにバンド幅が低下する．小規模システムで

表2 Cenju-3における予測モデルのパラメータ

Table 2 Parameter in estimation model of Cenju-3.

CPU		V_R4400SC 75 MHz
基準速度 V_{ref}		23.13（MFLOPS）
1ステップの計算時間 a	a_{hit1}	0.25（μs）
	a_{hit2}	0.27（μs）
	a_{miss}	0.36（μs）
キャッシュサイズ		1次16K，2次1M
通信バンド幅 b		式(9)参照

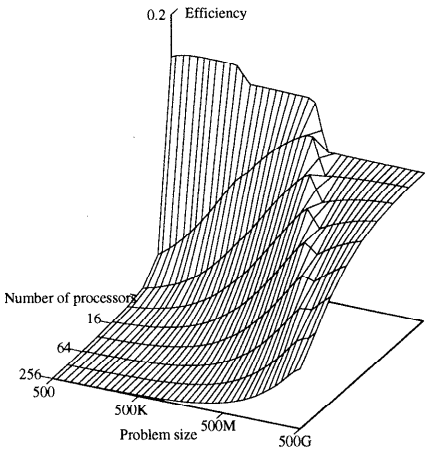


図11 行列乗算プログラムの予測効率曲面（Cenju-3）．1次キャッシュの影響はほとんど見られない．

Fig. 11 Estimated efficiency surface for matrix multiplying on Cenju-3. It shows the 1st cache memory has little effects on the efficiency.

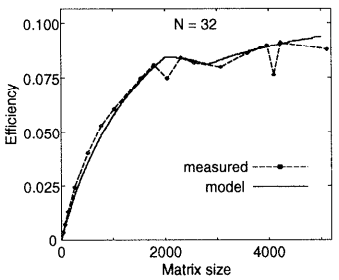


図12 Cenju-3のモデル予測値と実測値との比較（$N = 32$における断面図）．

Fig. 12 A comparison of an estimated performance and a measured performance at $N = 32$.

の実測の結果，

$$b^{-1} = 0.015 - \frac{0.0238}{N^{1.5}} \quad (\mathrm{ms/B}) \tag{9}$$

という関係があることが分かった．

図11に予測した効率曲面，**図12**には実測値との比較のためにモデルから推測した効率曲面と実際にすべての点で測定して求めた効率曲面の各々の$N = 32$における断面図を記す．やはり挙動がよくとらえられ

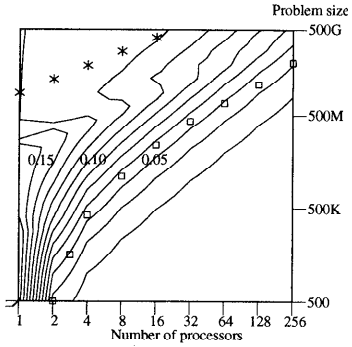


図 13 Cenju-3 の効率曲面の等高線表示．＊印はメモリ容量による計算の限界点，□印は AP1000 と性能が逆転する点を表す．等高線の傾きが AP1000 より大きい．

Fig. 13 Contour map of the efficiency surface of Cenju-3. *s describe a computation limit by real memory capacity. □s describe a point where AP1000's performance overcomes Cenju-3's one.

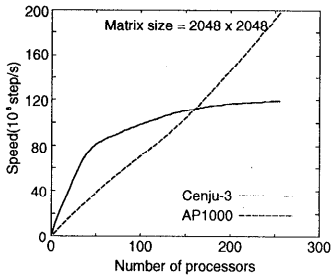


図 14 AP1000 と Cenju-3 の処理速度が逆転する様子

Fig. 14 The AP1000's processing speed overcomes Cenju-3's when a large number of PE are used.

ていることが示されている．また，効率曲面を等高線形式で表示したものを**図 13** に示す．等高線の傾きが AP1000 と比べて大きく，プロセッサ数を増やしたときの効率低下の度合が大きいことが分かる．そのため，**図 14** に示すように，プロセッサ単体の処理速度は Cenju-3 の方が速いが，プロセッサ数の増加とともに AP1000 の性能と逆転する現象が見られる．2048 × 2048 行列の場合，約 160 台を超えると性能が逆転し，AP1000 の方が速くなった．図 13 中□印は性能が逆転する点を表しており，この点列より右下の領域では，AP1000 の方が処理速度が速くなる．

(d) ワークステーション・クラスタ

SPARCstation 10 で構成された PVM による WS クラスタでの効率曲面を予測モデルから求めた．プロセッサ 1〜16 の小規模システムで実測したパラメータを**表 3** に記す．相互結合網は単一バスと見なせるため，N の増加とともにバンド幅が低下する．実測によれば，メッセージサイズの増加に対してもバンド幅が低下することが分かり，

表 3 WS クラスタにおける予測モデルのパラメータ

Table 3 Parameters in estimation model of WS cluster.

CPU		SuperSPARC 40 MHz
基準速度 V_{ref}		19.69 (MFLOPS)
1 ステップの計算時間 a	a_{hit}	0.247 (μs)
	a_{miss}	0.260 (μs)
キャッシュサイズ		16 (KB)
通信バンド幅 b		式 (10) 参照

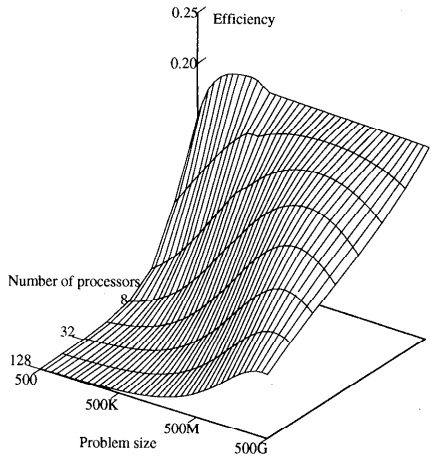


図 15 WS クラスタでの予測効率曲面．問題サイズが増加しすぎると逆に性能が下がる．

Fig. 15 Estimated efficiency surface of WS cluster. When the problem size becomes too large, the efficiency goes lower.

$$b^{-1} = \beta\left(1 + \alpha\frac{n^2}{N}\right) \tag{10}$$

というモデルをたてた．ただし，α はパケット作成に要するコストの増加率，β は N により定まる最小通信コストである．小規模システムでの実測の結果，

$$\alpha = 4.5 \times 10^{-7}$$

$$\beta = 3.5 \times (\log N)^{0.5}\ (\mu\mathrm{s/B})$$

となった．

予測モデルから求めた効率曲面を**図 15** に示す．予測モデルの検証のために，予測モデルを用いて求めた効率曲面と実際に測定して求めた効率曲面の各々の $N = 16$ における断面図を**図 16** に示す．また，曲面を等高線形式で表示したものを**図 17** に示す．キャッシュの影響がほとんどなく，ほぼ単調な曲面になることが分かる．

5.2 アルゴリズム B との比較

ここでは，異なるアルゴリズムの性能予測と性能比較をした例を紹介する．

(a) アルゴリズム B の予測モデル

アルゴリズム B で行列乗算を行うと，計算時間 T_c

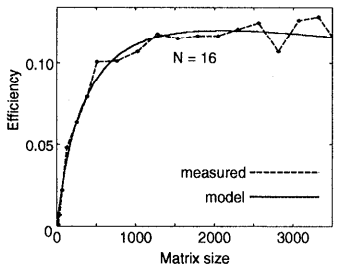


図16 WS クラスタのモデル予測値と実測値との比較（$N = 16$ における断面図）

Fig. 16 A comparison of an estimated performance and a measured performance at $N = 16$.

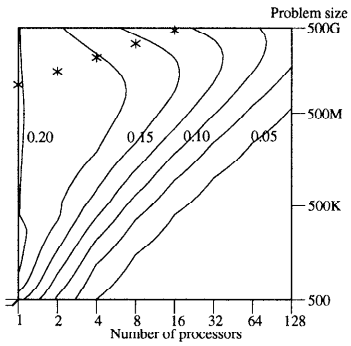


図17 WS クラスタの効率曲面の等高線表示．＊印はメモリ容量による計算の限界点を表す．

Fig. 17 Contour map of efficiency surface of WS cluster. *s describe a computation limit by real memory capacity.

は各 PE での演算回数から，

$$T_c = a \times \frac{n^3}{N\sqrt{N}} \times \sqrt{N} = an^3/N \qquad (11)$$

となる．一方，オーバヘッド T_o には通信オーバヘッドのほかに PE 間での集計の際の加算も含めることになる．加算 1 演算に要する処理時間を a' とすると，

$$T_o = \left(a' + \frac{2}{b}\right) \times \frac{n^2}{N} \times \sqrt{N}\left(\sqrt{N} - 1\right)$$
$$= \beta\left(1 - \frac{1}{\sqrt{N}}\right) n^2$$

となる．ただし，$\beta = a' + \frac{2}{b}$ とした．

(b) 予測効率曲面と性能比較

プロセッサ数 1～8 の小規模システムで実測した予測モデルのパラメータを**表 4** に記す．メモリアクセスの順序などが異なるために，表 1 のアルゴリズム A に対するものとは若干異なる値となる．予測モデルを用いて効率曲面の形状を推測し，アルゴリズム A と比較した様子を**図 18** に示す．両者はまったく異なるアルゴリズムだが，結果的にほぼ同じ挙動をとること

表4 アルゴリズム B の予測モデルのパラメータ

Table 4 Parameters in estimation model of matrix multiplying by the algorithm B.

1 ステップの計算時間 a	a_{hit}	0.70 (μs)
	a_{miss}	1.10 (μs)
オーバヘッド部分の係数 β		5.8 (μs)

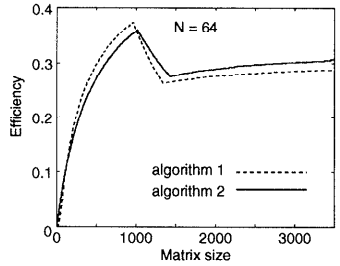


図18 AP1000 上での 2 つのアルゴリズムの性能比較（$N = 64$ における断面図）

Fig. 18 A comparison of algorithm A and B on AP1000 (Cross section of the efficiency surface at $N = 64$).

が分かった．

6. ま と め

並列計算機を効率的に利用し，高い性能を引き出すためには，その性能の全体的な挙動をつかむ必要がある．ところが，これまでの性能評価手法では，ある特定の利用形態での性能を知ることはできても，全体の性能を知るには不十分であった．

本稿では，性能の全体的な挙動を，1 つの数値や数式で表現するのではなく，性能を決定づける個々の要因ごとの簡単なモデルからボトムアップ的に予測する方法を提案した．要因を的確に把握し，パラメータを適切に与えれば，簡単なモデルでも十分実用的なレベルで性能予測が行えることを示した．また，こうして予測した全体挙動から，極大性能となる利用形態が分かり効率的に実行するための指針が得られるなどの有用性を示した．

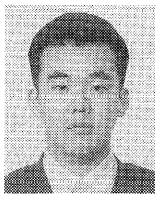
本稿では 3 種類の異なるアーキテクチャでの予測結果を示したが，いずれも分散メモリ型という点では共通である．これ以外のアーキテクチャでの有効性を明確にすることは今後の研究課題である．

参 考 文 献

1) 関口智嗣，佐藤三久：並列システムのスケーラビリティによる評価，情報処理学会研究報告，ARC-111-3, pp.17–24 (1995).
2) 関口智嗣，佐藤三久，井須芳美，長嶋雲兵：定量的な並列システムのスケーラビリティ評価指標，

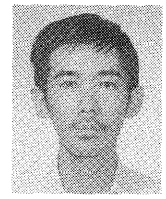
情報処理学会並列処理シンポジウム '96, pp.235–242 (1996).
3) Amdahl, G.: Validity of the Single-processor Approach to Achieving Large Scale Computing Capabilities, *Proc. AFIPS Conf.*, pp.483–485 (1967).
4) Gustafson, J.: Reevaluating Amdahl's Law, *CACM*, Vol.31, pp.523–533 (1988).
5) Sun, X.-H. and Ni, L.M.: Scalable Problem and Memory-bounded Speedup, *J. Parallel Distrib. Computing*, Vol.9, pp.27–37 (1993).
6) Kumar, V. and Singh, V.: Scalability of Parallel Algorithms for the All-pairs Shortest Path Problems, *Proc. Conf. on Parallel Processing*, pp.136–140 (1990).
7) Sun, X.-H. and Rover, D.T.: Scalability of Parallel Algorithm-machine Combinations, *IEEE Trans. Parallel and Distributed Systems*, Vol.5, No.6, pp.599–613 (1994).

(平成 8 年 9 月 13 日受付)
(平成 9 年 1 月 10 日採録)



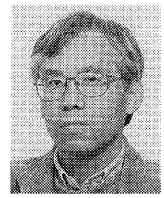
古市 実裕（正会員）

1994 年東京大学工学部計数工学科卒業，1996 年同大学大学院修士課程修了．同年より日本アイ・ビー・エム（株）東京基礎研究所に勤務．現在，コンピュータ・アーキテクチャ，システム・ソフトウェア，ユーザ・インタフェースなどの研究に従事．計測自動制御学会会員．



永松 礼夫（正会員）

1980 年東京大学工学部計数工学科卒業，1982 年同大学大学院修士課程修了．1984 年同博士課程退学，同年より東京大学工学部助手を経て，1995 年より会津大学情報センター助教授，現在に至る．並列処理アーキテクチャ，並列実行管理システム，画像処理の研究に従事．日本ソフトウエア科学会，計測自動制御学会，IEEE，ACM などの会員．



出口光一郎（正会員）

1976 年，東京大学大学院修士課程修了（計数工学）．同年より東京大学工学部助手，講師を経て，1984 年，山形大学工学部情報工学科助教授，1988 年，東京大学工学部計数工学科助教授，現在に至る．この間，1991～1992 年，米国ワシントン大学客員準教授．コンピュータビジョン，画像計測，並列コンピュータの研究に従事．計測自動制御学会，電子情報通信学会，形の科学会，IEEE などの会員．