Vol.2009-HPC-121 No.28
2009/8/6

高並列処理における並列性能評価方法

折居茂夫†

測定から決定した並列効率と処理時間モデルをフィッティングするモデル化方法と，そのモデルパラメータを用いた並列性能評価方法を提案する．このモデル化方法により，並列処理部の処理時間として測定した中に紛れ込んだ並列オーバーヘッドを検出することが可能となる．また少ない測定値を基にした並列性能予測が可能となる．

Method of Parallel Performance Evaluation for Highly Parallel Processing

Shigeo Orii†

The author has proposed a modeling method which fits between a time model of parallel program and measured timing data of the program through a parallel efficiency metric. The author has also shown a parallel performance evaluation method based on the model parameters. By using this modeling method, one can detect parallel overhead time hiding in timing data measured as parallel processing parts of the program and also predict parallel performance with a few points of measured data.

1. はじめに

ペタフロプスの計算能力を持つ数万台規模のプロセッサを持つ計算機が稼動する現在，高並列処理の並列性能評価方法の研究は，巨大な資源の有効利用という観点から益々重要になる．

そこで高並列処理に適応できる並列効率のメトリックを提案した[1]．逐次時間を用いないこの並列効率のメトリックにより，高並列処理を行う任意の計算機とアプリケーションプログラムから成るシステムの性能を効率で評価することが可能となった．

一方さらに詳細な性能評価を行うためには，並列処理時間に対してプロセッサ数と問題の大きさを関係づけることが必要となる．それには処理のアルゴリズムをプロセッサ数と問題の大きさを表す変数でモデル化する方法が古くから存在し，改良が重ねられている[2]．一方この方法は処理が複雑になるとモデルが巨大で複雑になり，任意のシステムに適用できない難点がある．また実際の並列オーバーヘッドを測定すると，その処理時間はプロセッサ数の変化に対して複雑な挙動を示し，モデルパラメータを決定することは容易でない[3]．

そこで今回は，なるべく多くのシステムに適用できる並列処理時間のモデル化方法と，モデルパラメータを用いた性能分析と予測方法を提案する．

2 章では本研究で使用する並列効率メトリックを説明する．3 章では測定値から求めた並列効率メトリックを用いた並列処理時間のモデル化方法を提案する．4 章では富士通 PRIMEPOWER HPC2500 と High Performance Linpack[4]から成るシステムのモデル化を行い，それらの性能分析と性能予測を論じる．

2. 並列効率メトリックとその可視化

並列処理における並列効率は一般に式(1)のように定義される．

$$E_p(p,n) \equiv \frac{\tau(1,n)}{p \cdot \tau(p,n)} \tag{1}$$

ここに$\tau(p,n)$は並列処理時間であり，p はプロセッサ数，n は問題の大きさである．この定義は基準が$\tau(1,n)$と明確である半面，高並列処理において$\tau(1,n)$を測定できないという問題をもつ．これを解決するため，高並列処理の並列効率を記述する，$\tau(1,n)$を用いない並列効率メトリックを式(2)として導いた[1]．

†富士通株式会社
Fujitsu Limited

Vol.2009-HPC-121 No.28
2009/8/6

$$\varepsilon_p(p,n) \equiv \frac{\gamma(p,n)}{\tau(p,n)} = 1 - \frac{\chi(p,n)}{\tau(p,n)} \tag{2}$$

ここに $\gamma(p,n)$は並列計算部，$\chi(p,n)$は並列オーバーヘッド部の処理時間，$\tau(p,n)=\gamma(p,n)+\chi(p,n)$である．この並列効率メトリックは基準が従来の並列効率の$\tau(1,n)/p$ から $\gamma(p,n)$に代わったものである．この$\varepsilon_p(p,n)$を用いれば高並列処理の効率の評価が可能となる．

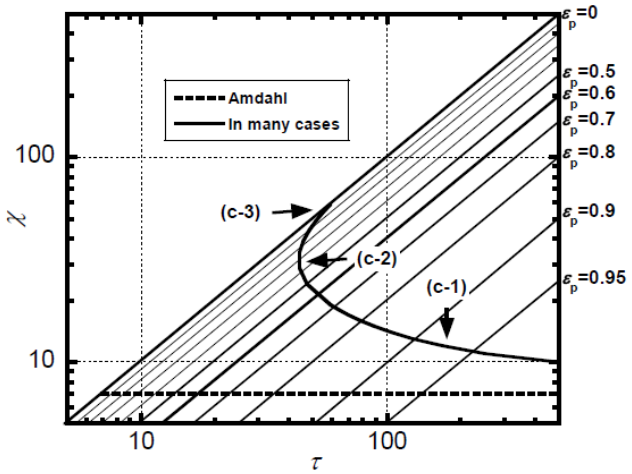

図 1　処理時間τ_i，並列オーバーヘッドχ_i，並列効率ε_p の同時可視化　（τ-χ空間）

式(2)で表されたこの効率は，横軸を$\tau(p,n)$，縦軸を$\chi(p,n)$とした座標系で直線になる．図 1 に概念図を示す．図 1 において処理時間は下三角部分に記述され，対角線は$\tau(p,n)=\chi(p,n)$で，並列処理でこれ以上時間短縮できない限界時間を示す．図中の破線は$\chi(n)\neq 0$ の場合で，並列オーバーヘッドχが p と共に増加しない Amdahl の法則がこれに当たる．実際の並列処理は$\chi(p,n) \neq 0$ であり，(c-2)のように対角線に至る前に処理時間が最小値を迎える．(c-1)はプロセッサの増加と共に処理時間$\tau(p,n)$が減少する領域，(c-3)はプロセッサの増加と共に処理時間が増加する領域である．

図 1 の対角線とそれに平行なラインは並列効率が等しい Isoefficiency ラインである．式(2)が示すように，この座標系のスケールが決まるとこのラインは一義的に決まる．結果，この座標系に 1 点をプロットすると，それに対する処理時間，並列オーバーヘッド及び並列効率が可視化され，並列処理の特性を知ることができる．そこで本論文では以後，この座標系を τ - χ 空間と呼ぶことにする．

3.　高並列処理における並列性能評価方法

測定より決定した並列効率メトリック$\varepsilon_p(p,n)$と処理時間モデルをフィッティングし，モデルパラメータを求める．実際の測定値を扱うため，(1)ロードインバランスを考慮した並列効率メトリックを定義し，フィッティング式を導出する．次に(2) 処理時間モデルを導入し，(3) 並列処理部にある並列オーバーヘッドを考慮してモデルパラメータ決定をする．

3.1　処理時間のモデル化方法

(1)　ロードインバランスを考慮した並列効率メトリックとフィッティング式の導出

実際の並列処理において生じるロードインバランスを考慮するため，処理時間の関係を式(3)のように拡張すると式(2)は式(4)となる．

$$p \cdot \tau(p,n) \equiv \sum_{i=1}^{p} \gamma_i(p,n) + p \cdot \chi(p,n) \tag{3}$$

$$\varepsilon_p(p,n) \equiv \frac{\sum_{i=1}^{p} \gamma_i(p,n)}{p \cdot \tau(p,n)} \tag{4}$$

測定値から求めた並列効率メトリックのみでフィッティングできるよう，式(4)を式(5)のように変形し，これをフィッティング式とする．

$$\frac{1-\varepsilon_p(p,n)}{\varepsilon_p(p,n)} = \left(p \cdot \tau - \sum_{i=1}^{p} \gamma_i(p,n) \right) / \sum_{i=1}^{p} \gamma_i(p,n) \tag{5}$$

式(5)の左辺は並列効率メトリックのみで構成されている．効率の観点から例えば$\varepsilon_p(p)$<0.1 でシステムを使うことはないと考えると，式(5)により右辺の現象を 0.1<$\varepsilon_p(p)$<1 の範囲に限定してモデル化できる．並列処理時間をモデル化する場合，全ての測定データに当てはまるモデルを作ることは難しいが，この範囲で有意なモデルの作成は比較的容易である．またモデル化に利用する測定データもこの範囲で取捨選択できる．

並列オーバーヘッドの処理時間χをモデル化する場合，通常χの測定値は不規則な変動を伴い，この変動がモデルパラメータ決定に大きく影響を与える場合がある．このような場合，そのモデルは性能分析や予測には不向きと考えられる．これに対し式(5)の左辺は$\chi/\Sigma\gamma/p$ ゆえ，χの小さな変動は大きな$\Sigma\gamma/p$ の下で小さく扱うことができる．このように並列処理時間のモデル化に必要のない変動を抑制することにより，右辺の並列オーバーヘッドモデルの簡略化が可能となる．

これらのことから式(5)で多くのシステムの処理時間をモデル化できると考える．

(2) 処理時間モデルの導入

問題の大きさを一定として処理時間モデルの式(6)をフィッティング式(5)に代入すると式(7)を得る．

$$\tau(p) \equiv \frac{a}{p} + \chi(p) \tag{6}$$

$$\frac{1-\varepsilon_p(p)}{\varepsilon_p(p)} = \frac{a + p\cdot\chi(p)}{\sum_{i=1}^{p}\gamma_i(p)} - 1 \tag{7}$$

式(7)においてpを横軸にとり左辺を測定した処理時間から計算して縦軸にプロットすると，並列オーバーヘッド$\chi(p)$はそのグラフの傾きとなる．したがってプロセッサ数に依存しない逐次処理時間は，このグラフの 1 次の直線の傾きとなる．

(3) 並列処理部に含まれる並列オーバーヘッドの考慮

処理時間測定において，並列処理部の処理時間として測定した$\gamma_i(p)$にもプロセッサの増加とともに減少しない部分と増加する部分即ち，並列オーバーヘッドが含まれる可能性がある．そこで$\varepsilon_p(p)$の算出において，あるプロセッサ数 $p=p_1$ に固定した $\gamma_i(p_1)$ を用いる．これを式(8)に示す．

$$\varepsilon'_p(p) \equiv \frac{\sum_{i=1}^{p}\gamma_i(p_1)}{p\cdot\tau(p)} \tag{8}$$

$\gamma_i(p_1)$のように固定した結果，$p_1<p$ において並列処理部分に並列オーバーヘッドの増減が，並列効率$\varepsilon_p'(p)$の値に反映される．この並列オーバーヘッドを$\chi'(p)$と置くと，式(7)は式(9)となる．

$$\frac{1-\varepsilon'_p(p)}{\varepsilon'_p(p)} = \frac{a + p\cdot\chi'(p)}{\sum_{i=1}^{p}\gamma_i(p_1)} - 1 \tag{9}$$

ここで右辺を多項式で近似すると式(10)となる．

$$\frac{1-\varepsilon'_p(p)}{\varepsilon'_p(p)} = \sum_{i=0}^{i_{\max}} c_i \cdot p^i \tag{10}$$

多項式の各係数は式(9)に対し，$c_0 = a/\sum_{i=1}^{i_{\max}}\gamma_i(p_1) - 1$，$\sum_{i=1}^{i_{\max}} c_i\cdot p^i = p\cdot\chi'(p)/\sum_{i=1}^{p}\gamma_i(p_1)$ の関係となる．

ここで 2 次近似とし，$i_{\max}$=2 とすると式(11)を得る．

$$\frac{1-\varepsilon'_p(p)}{\varepsilon'_p(p)} \approx c_0 + c_1\cdot p + c_2\cdot p^2 \tag{11}$$

式(11)は，逐次処理時間にようにプロセッサ数に関係していない並列オーバーヘッドも，1 次の傾きとして検出できることを示す．

多項式の係数とモデルパラメータの関係は式(12)，(13)となる．

$$a = \sum_{i=1}^{p}\gamma_i(p_1)\cdot(1+c_0) \tag{12}$$

$$\chi'(p) = \sum_{i=1}^{p}\gamma_i(p_1)\cdot(c_1 + c_2\cdot p) \tag{13}$$

p を変えて$\tau(p,n)$，$\gamma_i(p,n)$を測定し，式(8)で$\varepsilon'_p(p,n)$を求めて式(11)でフィッティングすると c_0，c_1，c_2 が得られ，モデルパラメータ式(12)と，モデルパラメータから成る並列オーバーヘッドモデル式(13) を決定することができる．

式(11)の多項式の係数は処理時間の性質から有効範囲が存在する．第 1 に式(12)は $1+c_0 \geqq 0$ である．第 2 に $c_1 \geqq 0$ である．また多くの場合 $c_2 \geqq 0$ である．フィッティング結果がこの有効範囲外の場合は，その原因を一考する必要がある．

式(6)のχに式(13)のχ'と式(12)を代入し，両辺を p で微分したものをゼロと置くと，並列処理による最小処理時間 $\tau_{\min}$ におけるプロセッサ数 p_c を式(14)として求めることができる．

$$p_c = \sqrt{\frac{1+c_0}{c_2}} \tag{14}$$

3.2 並列性能評価方法

性能評価として，始めに図 1 を用いた(1)定性的性能分析を行い，さらに詳しい分析

が必要な現象に着眼する．次に 3.1 節で示した実測値とモデルのフィッティングによる(2)モデルパラメータ決定を行い，それを用いて(3)定量的な性能分析を行う．今回評価したアプリケーションプログラムと計算機からなるシステムは，High Performance Linpack（以下，HPL）[5]と富士通 PRIMEPOWER HPC2500 である．HPL が flop/s の計算に使っている処理時間をτ_iとし，dgemm，dtrsm，dtrsv の処理時間の総和を並列処理部の処理時間γ_iとし，これらをプロセッサ毎に測定した．主な入力データは，問題のサイズ N=2000，5000，10000，20000，ブロックサイズ NB=10，プロセスグリッド P × Q は P=1 とし Q を変えてプロセス数をコントロールした．

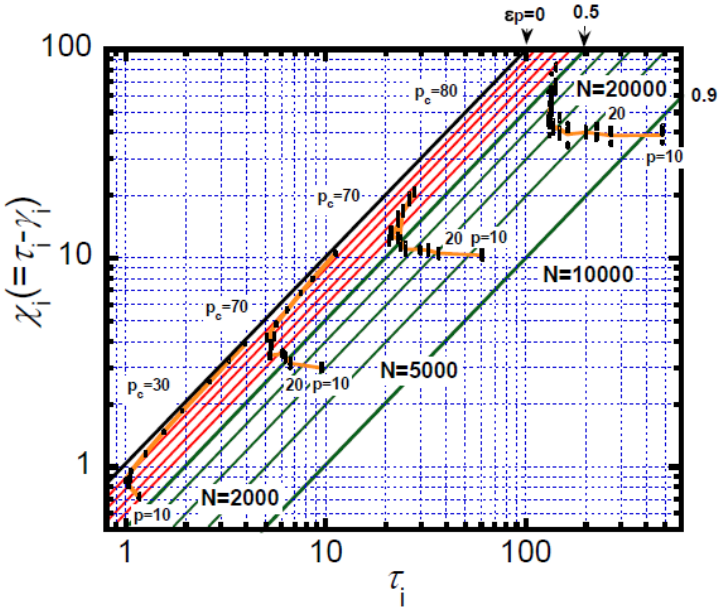

図２　「HPL＆HPC2500」システムにおける全プロセッサの性能特性の可視化
（τ_i：処理時間，χ_i：並列オーバーヘッド，ε_p：並列効率）

(1) 定性的性能分析

図２は全プロセッサの測定値によるシステムの状態をτ-χ空間で可視化したものである．オレンジ色の線はχ_iの各プロセッサの平均値を表す．平均値からの縦のずれからシステムがロードインバランスを起こしていることがわかる．図はχ_iの平均値からの偏差が N の増加とともに増大することを示す．一方τ_iの偏差が殆どないことを示す．したがってこのロードインバランスはτ_iの測定だけでは検出できないことがわかる．

N=20000 について Isoefficiency ラインをみると，p=40 においてχ_iの縦のずれを原因としたε_pの最大偏差が約 0.1 に達していることがわかる．この幅はプロセッサ増加とともに増えている．さらに N=20000 と他の N と比較すると，他の N が滑らかなカーブで最小処理時間となるのに対し，最小処理時間に近づくに従いχのばらつきが大きくなる．したがってこの最小処理時間は，ロードインバランスの影響を受けているように見える．

図２はN=20000にの p=10 において，ε_p=0 の時点の処理時間は 40 秒である．故に実際の最小処理時間τ_{min}は$10<\tau_{min}<500(=\tau_p(10))$にあるが，$\tau_{min}$の実測値は，約 130 である．

(2) モデルパラメータ決定

τ-χ空間で観測された事象をモデル化するため，測定値τ_iとγ_iを用いε'_pを計算し，式(11)とフィッティングした．その結果得られた多項式の係数 $\boldsymbol{c_0}$, $\boldsymbol{c_1}$, $\boldsymbol{c_2}$ を表 1-1 に示す．ここに **R** は相関係数である．

表 1-1　システム（HPL＆HPC2500）のフィッティング結果

N	p_1	c_0	c_1	c_2	R
2000	10	-0.0543	0.153	0.00147	0.999
5000	10	0.421	0.0168	0.000670	0.9938
10000	10	0.303	0.00144	0.000351	0.9984
20000	10	0.0810	0.00275	0.000164	0.9995

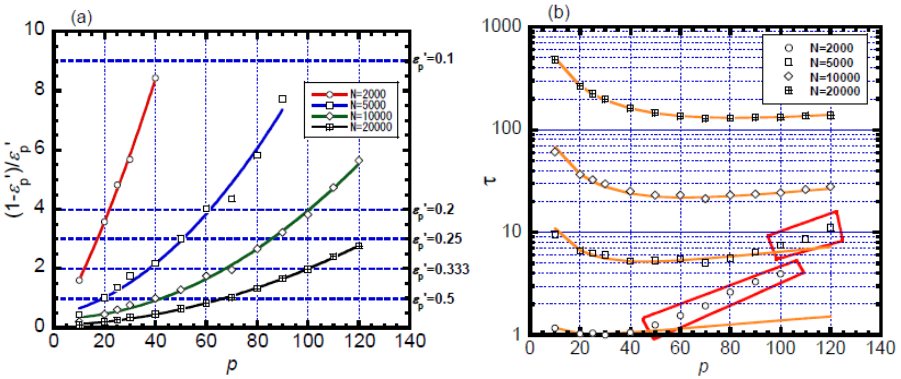

図３　「HPL＆HPC2500」システムのモデルパラメータ決定
(a) 測定値と式(11)のフィッティング結果．(b) 測定値τ（マーク）とモデル曲線の比較

表 1-1 の多項式の係数で式(11)の右辺を計算し，測定値から計算した左辺との比較を図 3 (a)に示す．0.1<$\varepsilon_p(p)$ <1 の範囲で両者がよく一致していることがわかる．

次に多項式の係数と式(12), (13)からモデルパラメータ a と並列オーバーヘッドモデルχ'を求めたものを表 1-2 に示す．これらを式(6)に代入して処理時間を求め，実測値と比較したものを図 3 (b)に示す．図中赤で囲んだ測定値は，ε_p'<0.1 であったためフィッティングに使用しなかった測定値で，これらを除けばモデルと実測値はよく一致していることがわかる．

表 1-2　システム（HPL＆HPC2500）の測定値とモデル値

	測定				モデル				
N	$\Sigma\gamma_i(10)$	τ_{min}	p_c	$\varepsilon_p(p_c)$	a	χ'	p_c	τ_{min}	$\varepsilon'_p(p_c)$
2000	4.494	1.001	30	0.136	4.25	0.688 + 0.00661･p	25.4	1.02	0.164
5000	66.15	5.068	70	0.178	94.0	1.11+0.0443･p	46.1	5.19	0.393
10000	503.3	21.27	70	0.393	656	0.725+0.177･p	60.9	22.3	0.483
20000	4445	130.7	80	0.629	4805	12.2+0.729･p	81.2	131	0.453

(3) 定量的性能分析

表 1–2のモデルパラメータを用いて性能を分析することができる．N=2000 の a を基準とすると，問題の大きさが N=2000，5000，10000，20000 と大きくなるに従い，処理時間は 1，　22.1，　154，1131 倍と 1 オーダーずつ大きくなる．

ここで(1)の定性的分析で着目したN=20000の性能を調べる．χ'の第 1 項は N=2000，5000，10000 ではほぼ 1 なのに対し，N=20000 では 12 と 10 倍大きな値となる．

図 5 はモデルより求めたχ'と測定より求めたχの比較である．(a)，(b)，(c)はχ'とχの差は少ないが，N=20000の(d)ではプロセッサ数の増加と共にその差は広がり，p=120 ではχ'=100，χ=65 とその差が 35 と大きくなることが分かる．モデル式をみると，この増分のうちの 12.1(=12.2-0.1)は，χ'の増加がもたらしたもの，残りは第 2 項の増加によりもたらされたものと考えられる．この増加は，3.2 節の(1)で述べたロードインバランスのためと推測される．もしそうであればロードインバランスを解消することにより現状のτ_{min}=131 は，96 になる可能性がある．

表 1-2 の測定とモデルのτ_{min}を比較するとよく一致している．一方 p_c は N により一致する場合とそうでない場合がある．図 3 の(b)をみると，この不一致はプロセッサ数の増加に対して処理時間τの変動が少ないためであり，実測値から求めたτ_{min}の p_c が微妙な変動により決まるためであることがわかる．

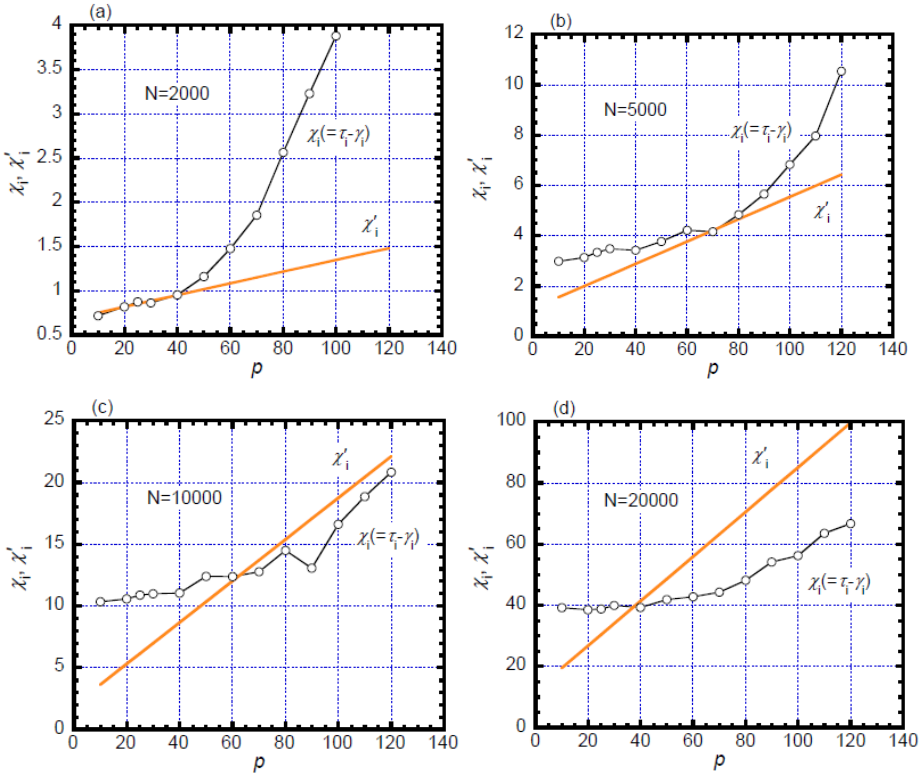

図 5　「HPL＆HPC2500」システムの並列オーバーヘッド
（χ_i：測定値．χ'_i：モデルパラメータ）

表 1-3　Isoefficiency のプロセッサ数

N	**p**			
	(ε'_p=0.9)	**(ε'_p=0.5)**	**(ε'_p=0.25)**	**(ε'_p=0.2)**
2000	0.7	5.8	16.1	20.6
5000	7.3	35.2	63.2	80.4
10000	18.3	58.9	103.4	119.7
20000	20.0	73.2	132.5	154.2

表 1-3 にモデルから求めた Isoefficiency のプロセッサ数を示す．ε'_p=0.9 では N=2000 では逐次処理でもこの効率を得ることができないが，N=20000 では 20 台のプロセッサを投入することができる．ε'_p=0.5 では N=2000 で 6 台であったのが，N=20000 では 73 台を投入できる．

3.3 並列性能予測

ここでは式(11)を使った並列性能予測方法を示す．少ない実測値から予測を行うためε'_p (1)=1 と置き，これに実測値を加えたものを 3.2 で述べたようにフィッティングしてモデルパラメータを求める．これをモデル式に代入すると，プロセッサ数に対する処理時間が得られる．具体的な例として，図 6 (a)に N=20000 の場合の測定値 2 点から予測した場合と，図 6 (b)に 3 点から予測した場合を示す．丸印は測定値である．p_1 は予測に使った 1 つ目の測定値のプロセッサ数である．例えば図 6 (a)の p_1=20 のラインはこの点と p=30 の点とε'_p (1)=1 を使ってフィッティングして得たモデルの曲線である．図 6 (b)の p_1=40 のラインはこの点と p=50，60 の点とε'_p (1)=1 を使ってフィッティングして得たモデルの曲線である．図 6 は，p_1 を 20，40，60，80 と変化させた結果，これらが実測値と数十パーセントの差で一致することを示す．

図 6 のモデルパラメータと p_c を表 2-1，2-2 に示す． p_c の実測値は 80 ゆえ，表 2-2 の p_1=20 以外は大体一致している．

表 2-1 の p_1=80 と表 2-2 の p_1=60，80 は c_1 が負になっているので性能分析には適しないが，処理時間と p_c の予測には大きな影響を与えていない．

表 2-1　測定値 2 点とε'_p (1)=1 から予測したモデルパラメータと p_c　(N=20000)

p_1	$\Sigma\gamma_i(p_1)$	c_0	c_1	c_2	R	a	χ'	p_c
20	4568	-0.00514	0.00496	0.000187	1	4545	22.6+0.855･p	72.9
40	4935	-0.00116	0.000983	0.000175	1	4929	4.85+0.864･p	75.5
60	5583	-0.00123	0.00112	0.000110	1	5576	6.24+0.612･p	95.5
80	6522	0.00652	-0.00669	0.000175	1	6565	-43.7+1.14･p	75.9

表 2-2　測定値 3 点とε'_p (1)=1 から予測したモデルパラメータと p_c　(N=20000)

p_1	$\Sigma\gamma_i(p_1)$	c_0	c_1	c_2	R	a	χ'	p_c
20	4568	-0.0100	0.00801	7.4761e-5	0.9982	4522	36.6+ 0.342･p	115.1
40	4935	-0.00315	0.00259	0.000140	0.9997	4919	12.8+ 0.690･p	84.4
60	5583	0.00227	-0.00200	0.000157	0.9993	5596	-11.2+ 0.879･p	79.8
80	6522	0.00355	-0.00392	0.000142	0.9996	6545	-25.6+ 0.928･p	84.0

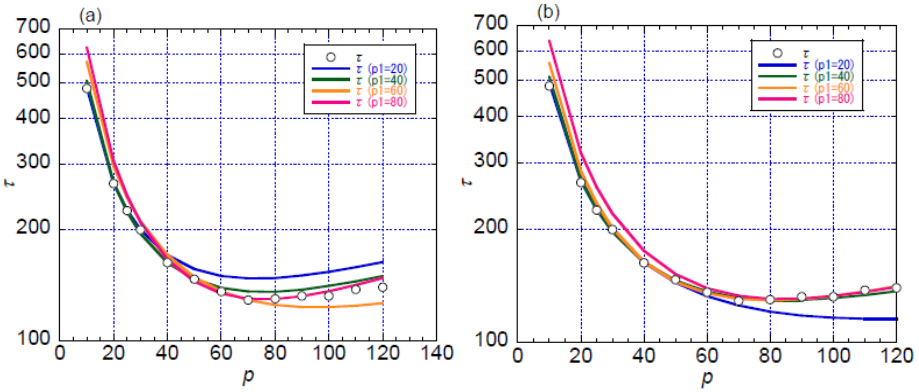

図 6　測定値に基づいた処理時間予測．使用測定点数：(a)：2 点，(b)：3 点．

4.　おわりに

計算機とアプリケーションプログラムシステムから成るシステム全体の処理時間の挙動を，並列効率を介して簡単な処理時間モデルに写し取る方法と，そのモデルパラメータを使った性能分析と性能予測方法を提案した. 並列効率を介することにより，並列オーバーヘッドモデルを簡略化できるこのモデル化方法は，多くのシステムに適用できると考える．

謝辞　富士通株式会社テクニカルコンピューティング･ソリューション事業本部の奥田基エグゼクティブアーキテクトの援助に感謝します．

参考文献

1) 折居茂夫 : 高並列処理におけるスケーラビリティ評価方法, 情報処理学会研究報告, 2009-HPC-119, Vol.2009, No.14, pp.169-174 (2009).
2) R. Susukita, et. al: Performance Prediction of Large-scale Parallel System and Application using Macro-level Simulation, In proceedings of SC2008.
3) 折居茂夫，レベル 1･2 並列ベンチマーク仕様及びそれに基づくスカラ並列計算機 SP2 のベンチマークテスト, JAERI-Data/Code 98−020 (1998).
4) A. Petitet, R. C. Whaley, J. Dongarra and A. Cleary: HPL - A Portable Implementation of the High-Performance Linpack Benchmark for Distributed-Memory Computers. http://www.netlib.org/benchmark/hpl/