社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2006－ARC－167 (17)
2006－HPC－105 (17)
2006／2／28

不均一クラスタの最適構成予測モデルの各応用への適用と評価

高橋 翔† 市川周一†

伝統的並列応用は負荷を PE に均等に分散するため，不均一クラスタ上では負荷不均衡により性能が低下する．高速 PE 上に複数のプロセスを起動すれば，応用を書き換えることなく性能向上を図ることができるが，最適なプロセス数やクラスタ構成を求めることは難しい．岸本と市川は，HPL を例として実行時間予測モデルを構築し，不均一クラスタの (準) 最適構成を予測できることを示した．本研究では，4 つの典型的科学技術応用 (CFD, FEM, FFT, HPL) に岸本の手法とその改良を適用し，(準) 最適な構成を予測することができるか検証する．測定結果によれば，改良された岸本の手法 (NP-T モデル) により，4 つの応用のいずれに対しても充分な精度で (準) 最適構成を予測することができた．

Application and Evalution of Optimal Configuration Estimation Scheme for Heterogenerous Clusters

Sho Takahashi † and Shuichi Ichikawa†

The performances of parallel applications are severly degraded on heterogeneous clusters by load-imbalance. Though it is simple to invoke multiple processes on fast PEs to alleviate load-imbalance, the optimal process allocation is not obvious. Kishimoto and Ichikawa constructed performance models for HPL (High Performance Linpack), with which the sub-optimal configurations of heterogeneous clusters were actually estimated. This study examines Kishimoto's scheme and its improvement on four typical scientific applications: CFD (computational fluid dynamics), FEM (finite element method), FFT (fast fourier transform) and HPL (high performance Linpack). The derived results indicated that an improved Kishimoto's model (NP-T model) works very well on any of these applications.

1. はじめに

既存の多くの並列応用では，要素計算機 (PE) の構成が均等であることを前提として，計算負荷を PE に均等に分散している．このような並列プログラムを不均一クラスタ上で実行すると，低速 PE がボトルネックとなって実効性能が低下する．不均一クラスタ上での性能を向上させるためには，PE の演算性能に応じて計算負荷を不均等に分散する必要がある．また，実行時間は通信時間にも影響されるので，全ての PE を使わず最適なサブセットを選択する方が高性能が得られる場合がある．最適な実行方法は，計算サイズによって異なる．

Kalinov ら[1] は，不均等な負荷分散を行う方法を 2 つに分類した．

HoHe 均一プロセス・不均一データ分散．各 PE 上で均一なプロセス数 (通常は各 1) を実行し，各プロセスの扱うデータ量を不均一にする．

HeHo 不均一プロセス・均一データ分散．各プロセスは均一なデータ量を扱うが，各 PE 上に不均一な数のプロセスを割り当てる．

ちなみに既存の並列応用は "均一プロセス・均一データ分散" なので，Kalinov の用語に従えば HoHo という表現になる．

HoHe は一般に性能面で有利であるといわれるが，既存応用を個別に不均一クラスタ用に書き換える必要がある．不均一クラスタを前提としたソース開発，デバッグ，性能チューニングを全ての応用に対して行うことは大きな負担である．一方 HeHo では，多くの場合 HoHo 型の応用を書き換えることなく，クラスタミドルウェアや並列処理環境で複数プロセスを生成することにより，不均一クラスタ上の性能向上を図ることができる．最適化に関していえば，HoHe で最適なデータ分割を求めることも，HeHo で最適なプロセス分割を求めることも，同様に難しい．

HoHe 型の負荷分散は多く研究されてきた．笹生ら[2] は，High Performance Linpack (HPL)[3] について HoHe と HeHo の性能比較を試みている．また，

† 豊橋技術科学大学 知識情報工学系
Department of Knowledge-based Information Engineering, Toyohashi University of Technology

笹生ら[4]は，NAS Parallel Benchmark のカーネルベンチマークについて，各 PE に割り当てるブロック数や分割幅を変える手法を実装している．大滝ら[5]は，$n \times n$ 行列積の Strassen アルゴリズムについて，再帰回数を考慮した領域分割を実装した．

HeHo 型の先行研究も幾つか存在する．Kalinov ら[1]は，超新星の三次元モデリングについて，実行時間を最小化するプロセス分散法を検討した．Cuenca ら[6]は，線形代数ルーチンに関して最適なプロセス分散法を検討した．岸本と市川[7][8]は，不均一クラスタ上で HPL についてマルチプロセス法 (HeHo) を検討し，実測値から実行時間予測モデルを構築した結果，実際に最適～準最適構成を予測できることを示した．上記の研究はいずれも 1 つの応用についてだけ HeHo 型負荷分散を検討しており，各手法が広範囲の応用に適用可能であるかは実証されていない．

本研究では，岸本と市川の手法[7][8]を 4 つのベンチマークプログラムに適用し，不均一クラスタ上の (準) 最適構成を求めることができるか検証する．さらに岸本らのモデルを改良して，より高精度な予測モデルが構築可能であることを示す．

2. 実行時間予測モデル

2.1 岸本らによる実行時間予測モデル

本節では，岸本らの予測モデルの概略を示す．詳細は文献 7)8) に記載されている．

N を問題のサイズとする．クラスタ内で等価な PE のグループ (サブクラスタ) を G_i であらわし，P_i は G_i 中で実際に使用する PE の数であるとする ($0 \le P_i \le |G_i|$)．G_i 内の PE には同数 (M_i 個) のプロセスを起動し，形式上 $P_i = 0$ のときは $M_i = 0$ であると定義する．クラスタ内の総プロセス数は，$P = \Sigma_i P_i M_i$ で表される．G_i の PE の実行時間を T_i と表すなら，我々の目標は T_i を N, P, M_i の関数として表すことである．全体の実行時間 T は $\max_i T_i$ で見積もることができる．この T の近似式を "実行時間予測モデル" とよぶ．可能な全ての構成についてモデルが構築できれば，それらのモデルを用いて最適な実行構成を予測することができる．

HPL のアルゴリズムと実装から，その実行時間は以下の式で表される．

$$T(N,P) = \frac{1}{P} \cdot O(N^3) + P \cdot O(N^2) + O(N^2) \quad (1)$$

上式より，ある P と M_i について，T_i は N の 3 次式で表せる (式 (2))．これを岸本らは N-T モデルと名づけた．式 (2) には 4 つの定数項があるが，これらの係数は実測値から最小二乗法で決定する．

$$T_i(N)|_{P,M_i} = k_0 N^3 + k_1 N^2 + k_2 N + k_3 \quad (2)$$

N-T モデルでは可能な全ての (P, M_i) の組に関してモデル構築が必要になるため，モデル数と実測時間が膨大になる．そこで複数の N-T モデルを統合し，P をパラメータとして含むモデルを構築する．このとき T_i は通信相手に依存せず P だけに依存すると仮定すると，各サブクラスタ内の測定だけでモデルが構築できるためモデル構築時間が削減できる．

$$T_i(N,P)|_{M_i} = k_4 P \cdot T_i(N)|_{P,M_i} + \frac{k_5}{P} \cdot T_i(N)|_{P,M_i} + k_6 \quad (3)$$

岸本らは式 (3) を P-T モデルと名づけた．P-T モデルでも係数は最小二乗法で決定するが，N-T モデルと違って全ての (P_i, M_i) について測定する必要はない．なお，$P = \exists M_i$(使用 PE 数が 1) の N-T モデルには PE 間通信時間が含まれないため，PE 数が 2 以上の N-T モデルから P-T モデルを構築する．

一度モデル構築が終われば，それらのモデルを用いて，与えられたサイズ N について最適な構成 (P_i, M_i) を予測することができる．具体的には，(1) 可能な全ての構成について，(2) モデルを用いて実行時間を予測し，(3) 実行時間を最小化する構成を発見して，(4) それを "予測最適構成" としてユーザに提示する．このような最適構成探索は一種の組合せ最適化問題であり，一般的には計算困難である．従って，非常に大規模な不均一クラスタの予測最適構成を探索する場合は，近似アルゴリズムを用いて探索空間を削減する必要がある．しかし 4 章で示すように，今回のような小規模な不均一クラスタでは，全数探索で最適構成を予測しても所要時間は 1 秒にも満たない．大規模クラスタのための最適構成探索アルゴリズムに関しては今後の課題である．

2.2 NP-T モデル

式 (1) から，N, P をパラメータとする T_i の近似式を直接導出すれば，一般に以下のようになる．

$$T_i(N,P)|_{M_i} = \frac{1}{P} \cdot (k_0 N^3 + k_1 N^2 + k_2 N + k_3) + P \cdot (k_4 N^2 + k_5 N + k_6) + k_7 N^2 + k_8 N + k_9 \quad (4)$$

この式は 10 個の係数 $k_0, ..., k_9$ を含むが，充分な数 (最低 10 個) の実測値があれば最小二乗法により係数を決定することができる．本研究では，こうして得られるモデルを NP-T モデルと呼ぶ．

NP-T モデルの用途は P-T モデルと同じであるが，式 (4) と式 (3) は等価ではない．それは，式 (3) に式 (2) を代入して，式 (4) と比較すれば自明である．P-T モデルは N-T モデルを統合するために導入された経験的モデルであり，NP-T モデルは計算量・通信量から自然に導出されるモデルであるといえる．

NP-T モデルは P-T モデルと同様に，各サブクラスタ内での実行時間だけを測定すればモデル構築が可能であり，P-T モデルと同じ測定データセットから構築できる．P-T モデルの測定データ数は，N の測定

表 1 測定環境

	サブクラスタ G_1	サブクラスタ G_2
CPU	Xeon 2.8 GHz	Celeron M 1.5 GHz
ノード数	8	8
OS	Redhat Linux 9	FedoraCore 3
Kernel	2.4.20-8	2.6.12-1.1381_FC3
Network	1000base-TX	1000base-TX
C	gcc 3.2.2	
Fortran	Intel Fortran Compiler 8.1	
MPI	mpich-1.2.6 (バッファサイズ 8KB)	

数 ×(PE 数-1) であり，充分に多い．

NP-T モデルは P-T モデルより未知係数が多く自由度が高いので，充分な数の測定からパラメータを抽出すれば，P-T モデルより正確なフィッテングが可能であると期待される．逆に考えると，NP-T モデルは自由度が大きい分，フィッティングの失敗によるモデル破綻が起こりやすい可能性もある．P-T モデルは単純化され未知変数が少ないため，比較的少ないデータからパラメータを抽出しても破綻しにくいかもしれない．このように利害得失があるので，結局 P-T と NP-T の実用上の優劣は，実応用と実システムを用いた評価結果に基づいて判断する必要がある．

3. 評価方法

3.1 評価環境と評価対象

多くのベンチマークプログラムでは計算サイズが固定されているが，性能予測に用いるためにはベンチマークはスケーラブルでなければならない[9]．本研究では，計算サイズが変更可能なベンチマークプログラム 4 種を実験に用いた．

Himeno BMT[10] 流体解析コードの非圧縮 Navier-Stokes ソルバの性能を評価するために開発されたプログラムで，圧力 Poisson 方程式を Jacobi 反復で解くカーネルの性能を測定する．

HPCMW[11] hpcmw-solver-test は，3 次元有限要素解析を各種アーキテクチャで評価するために開発されたベンチマークで，カーネル部分は並列 ICCG 法を実装している．

FFTE[12] サイズ $2^p \times 3^q \times 5^r$ の 1 次元／2 次元／3 次元離散フーリエ変換 (DFT) を行うプログラムで，ベンチマークでは倍精度複素 1 次元 DFT の性能を測定する．

HPL[3] 分散メモリ並列計算機用に Linpack ベンチマークを実装したもので，倍精度浮動小数点演算で線形代数方程式系を解く性能を測定する．本研究では，岸本ら[7][8] の結果を追試する意味で，HPL を測定対象に加えた．

評価に使用した不均一クラスタの構成を表 1 にまとめる．今回の実験では，ネットワークトポロジや帯域制限による問題の複雑化を避けるため，全ての PE を 1 つのスイッチ (ワイヤスピードスイッチ) で接続した．測定・評価の対象とするクラスタ構成 (P_i, M_i) と，その構成数を表 2 にまとめる．各ベンチマークの測定・評価範囲は，表 3 に示す．

表 2 不均一環境の構成

G_1	G_2	総組合せ数
$1 \le P_1 \le 8$	$0 \le P_2 \le 8$	144
$1 \le M_1 \le 2$	$0 \le M_2 \le 1$	

表 3 測定と評価に用いた問題サイズ N

	測定範囲	評価範囲
Himeno BMT	32〜192 (9 点)	32〜256 (10 点)
HPCMW	70〜504 (7 点)	70〜660 (20 点)
FFTE	2^{12}〜2^{20} (9 点)	2^{16}〜2^{23} (8 点)
HPL	400〜6400 (9 点)	1600〜9600 (7 点)

3.2 モデルの導出

Himeno BMT では，$N \times N \times N$ の問題を扱う．Himeno BMT の実行時間の内訳を解析したところ，ヤコビ反復法 $\frac{1}{P} \cdot O(N^3)$，領域間通信 $O(N^2)$，AllReduce 通信 $\log P \cdot O(1)$ であることがわかった．これらから，Himeno BMT の実行時間は以下のように見積もることができる．

$$T(N,P) = \frac{1}{P} \cdot O(N^3) + O(N^2) + \log P \cdot O(1) \tag{5}$$

実際には，通信時間のモデルはアルゴリズムだけでなく実装に依存する．例えば上の式では AllReduce を $O(\log P)$ で見積もっているが，AllReduce が逐次的アルゴリズムで実現される場合や，ネットワークスイッチの通信容量が小さくて通信が逐次化される場合は，AllReduce に $O(P)$ の時間が必要かもしれない．従って，モデル式を導出する際には，通信ライブラリの実装や，ハードウェア構成に注意する必要がある．

hpcmw-solver-test は本来 3 次元 $(N \times N \times N)$ の問題を扱うが，メモリ容量と実行時間の関係上，今回の実験では 2 次元 $(N \times N \times 1)$ の問題を測定した．hpcmw-solver-test の実行時間内訳は，行列ベクトル積 $\frac{1}{P} \cdot O(N^3)$，ベクトル内積 $\frac{1}{P} \cdot O(N^2)$，ベクトル加減算 $\frac{1}{P} \cdot O(N^2)$，領域間通信 $\frac{N^2}{P}$，AllReduce 通信 $\log P \cdot O(1)$ で抑えられる．これらから，HPCMW の実行時間は以下の式で見積もることができる．

$$T(N,P) = \frac{1}{P} \cdot O(N^3) + O(N^2) + \log P \cdot O(1) \tag{6}$$

実際には，今回の評価環境で $\log P$ の項を実測すると 100 ms 程度になり，$T(N,P)$ 全体からみて無視できる量となる．ところが実際に T の測定値から係数を抽出すると，$\log P$ の項に現実離れした係数が抽出されて，モデルの予測誤差を大きくする場合がある (パラメータ抽出における破綻の一例)．そのため，4 章で

表 4 Himeno BMT, HPCMW, FFTE の N-T, P-T, NP-T モデル

ベンチマーク	モデル	式		
Himeno BMT	N-T	$T_i(N)\|_{P,M_i}$	=	$k_0N^3+k_1N^2+k_2N+k_3$
	P-T	$T_i(N,P)\|_{M_i}$	=	$\frac{k_0}{P}\cdot T_i(N)\|_{P,M_i}+k_1+k_2\log P$
	NP-T	$T_i(N,P)\|_{M_i}$	=	$\frac{1}{P}\cdot(k_0N^3+k_1N^2+k_2N+k_3)+k_4N^2+k_5N+k_6+k_7\log P$
HPCMW	N-T	$T_i(N)\|_{P,M_i}$	=	$k_0N^3+k_1N^2+k_2N+k_3$
	P-T	$T_i(N,P)\|_{M_i}$	=	$\frac{k_0}{P}\cdot T_i(N)\|_{P,M_i}+k_1+k_2\log P$
			≈	$\frac{k_0}{P}\cdot T_i(N)\|_{P,M_i}+k_1$
	NP-T	$T_i(N,P)\|_{M_i}$	=	$\frac{1}{P}\cdot(k_0N^3+k_1N^2+k_2N+k_3)+k_4N^2+k_5N+k_6+k_7\log P$
			≈	$\frac{1}{P}\cdot(k_0N^3+k_1N^2+k_2N+k_3)+k_4N^2+k_5N+k_6$
FFTE	N-T	$T_i(N)\|_{P,M_i}$	=	$k_0N\log N+k_1N+k_2N^{\frac{1}{3}}+k_3$
	P-T	$T_i(N,P)\|_{M_i}$	=	$\frac{k_0}{P}\cdot T_i(N)\|_{P,M_i}+k_1P\cdot T_i(N)\|_{P,M_i}+k_2$
	NP-T	$T_i(N,P)\|_{M_i}$	=	$\frac{1}{P}(k_0N\log N+k_1N+k_2)+k_3P+k_4N+k_5N^{\frac{1}{3}}+k_6$

も示すとおり，最初から $\log P$ の項を省略したモデル式を構築してパラメータ抽出を行う方が，精度の高いモデルが得られる．このときの近似式は以下の通りとなる．

$$T(N,P)\approx\frac{1}{P}\cdot O(N^3)+O(N^2) \qquad (7)$$

FFTE の実行時間の内訳は，FFT 計算 $\frac{1}{P}\cdot O(N\log N)$，ベクトル分解 $O(N)$，テーブル作成 $O(N^{\frac{1}{3}})$，Alltoall 通信 $\frac{1}{P}\cdot O(N)+P\cdot O(1)$ で近似することができる．近似式は以下の通りである．

$$T(N,P)=\frac{1}{P}\cdot O(N\log N)+O(N)+O(N^{\frac{1}{3}})+\frac{1}{P}\cdot O(N)+P\cdot O(1) \qquad (8)$$

FFTE では，$N=2^n$ の計算を行う場合，プロセス数 P に 3 または 5 の因子が含まれると領域分割の方法が変わる．そのため，P が「3 または 5 の因子を含む場合」と「含まない場合」で，別なモデルを抽出して切り替えることにする．モデル式としては同じ式を用いるが，場合わけして別々に係数をフィッティングする．この場合わけは P-T モデルと NP-T モデルに対して行う．

以上の結果に従って，各ベンチマークの N-T, P-T, NP-T モデルの式を構築した (表 4)．

4. 評価結果

4.1 モデル構築時間と最適構成予測時間

各モデルを構築するための実測時間，実測データからパラメータを抽出する時間，そして与えられた N から最適構成を予測する時間を，表 5 にまとめる．NP-T モデルのための測定時間は P-T モデルと全く同一であるため，表では省略した．

表から明らかな通り，4 つの応用のいずれの場合でも，モデル構築 (パラメータ抽出) 時間は数秒以下で問題にならない程度である．与えられた N に対する最適構成予測時間は 0.2 秒以下なので，現状では応用の実行時間に対して充分小さいといえる．ただし，大規模な不均一クラスタでは選択可能な構成数が膨大になるため，最適構成予測時間が指数的に大きくなることが考えられる．そのような場合，近似アルゴリズムやヒューリスティックが必要になると思われるが，それは今後の課題とする．

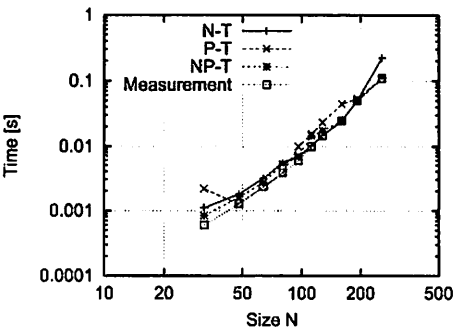


図 1 Himeno BMT

N-T モデルの構築に必要な測定時間は，P-T や NP-T モデルの数倍になる．本研究では，真の最適構成を求めるために，可能な全ての構成の実行時間を測定している．N-T モデルも可能な全ての構成について構築しているため，「N-T モデルのための測定時間」＝「真の最適構成を求める時間」である．

P-T モデルや NP-T モデルの予測精度が充分であるなら，実用上は P-T または NP-T モデルだけを構築して (準) 最適構成を予測すれば充分である．従って実用的な意味でのモデル構築時間は，P-T や NP-T の測定時間＋モデル構築時間となり，本研究で扱った 4 つの応用においては 10 秒〜5 時間となる．

4.2 予測最適構成の精度

本節では，予測された最適構成の実行時間が，真の最適構成の実行時間に対して，どれだけの精度をもつかを検討する．評価は，各モデルの予測最適構成の実測時間と，実測最適構成の実測時間を比較して行う．N が小さい範囲では，相対誤差が大きくとも実行時間差が小さいため問題は小さい．重要なのは，N の大きい範囲における予測誤差である．

N-T, P-T, NP-T モデルによる Himeno BMT の予

表 5 測定時間，モデル構築時間，最適構成予測時間 [sec.]

	N-T モデル			P-T モデル			NP-T モデル	
	測定	モデル構築	予測	測定	モデル構築	予測	モデル構築	予測
Himeno BMT	4.4×10^{1}	8.0×10^{-1}	4.6×10^{-2}	9.5×10^{0}	4.2×10^{0}	1.0×10^{-2}	1.6×10^{0}	1.0×10^{-2}
HPCMW	1.1×10^{5}	2.9×10^{0}	3.0×10^{-2}	1.8×10^{4}	1.5×10^{0}	2.0×10^{-2}	6.3×10^{-1}	2.0×10^{-2}
FFTE	3.8×10^{2}	2.3×10^{-1}	3.0×10^{-2}	4.3×10^{1}	2.0×10^{0}	1.9×10^{-1}	5.8×10^{-2}	1.0×10^{-2}
HPL	1.4×10^{4}	1.9×10^{0}	1.0×10^{-2}	2.9×10^{3}	2.0×10^{0}	5.0×10^{-2}	5.8×10^{-1}	1.0×10^{-2}

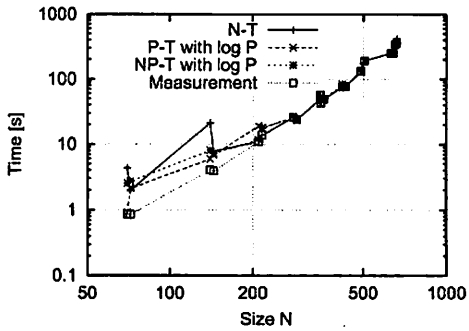


図 2 hpcmw-solver-test ($\log P$ の項がある場合)

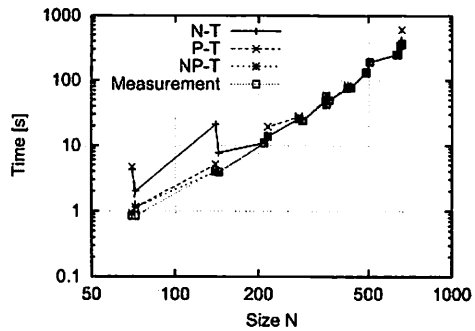


図 3 hpcmw-solver-test ($\log P$ の項を省いた場合)

測最良構成の (実測) 実行時間と，実測で最良の構成の (実測) 実行時間を図 1 に示す．図中，"Measurement" は実測最適構成の実測実行時間を示す．N-T モデルの予測誤差は，モデル内挿範囲の $96 \le N \le 192$ では最大 17%に収まっているが，外挿範囲である $N = 256$ では誤差が 100%を超えている．P-T モデルは $192 \le N \le 256$ で最大誤差 7%と準最適構成が予測できているが，$96 \le N \le 160$ では誤差が 50%を超えており精度は悪い．NP-T モデルは $160 \le N \le 256$ で最大誤差 3%に収まり，N-T，P-T モデルより高精度であることが確認できた．Himeno BMT については N-T モデルも P-T モデルも実用的でないが，NP-T モデルは充分な精度をもつといえる．

hpcmw-solver-test に関する結果を図 2 に示す．この図では，式 (6) の実行時間モデルに基づいて，$\log P$ の項を含んだ P-T モデルと NP-T モデルを構築している．一方図 3 では，式 (7) に示した近似を採用して，$\log P$ の項を省略した P-T モデルと NP-T モデルを評価している．図 2 と図 3 から明らかなように，$\log P$ の項を無視する方が正確なモデルが得られている．

図 3 では，$210 \le N \le 660$ で N-T モデルの誤差は最大 17%，NP-T の誤差は最大 9%と，どちらも (準) 最適構成が予測できている．P-T モデルは $280 \le N \le 650$ で誤差は最大 8%に収まるが，$N = 660$ では誤差 70%と増加する．hpcmw-solver-test では内挿，外挿共に N-T，NP-T モデルは実用的で，P-T モデルも内挿範囲では実用的と言える．

FFTE の評価結果を図 4 に示す．FFTE は通信時間が支配的であるという意味で，他の 3 つの応用とは大きく異なる．N-T モデルは $2^{18} \le N \le 2^{21}$ で誤差

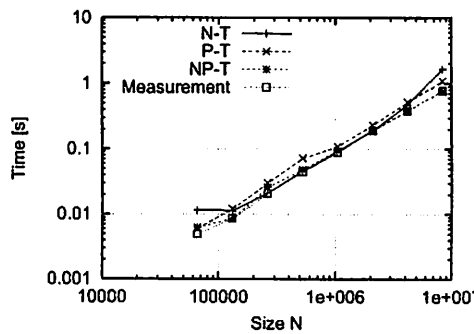


図 4 FFTE

は最大 3%に収まるが，外挿で N の増加につれ誤差が増大し $N = 2^{23}$ で誤差は 100%を超える．P-T モデルの誤差は $2^{16} \le N \le 2^{23}$ で 20〜60%となり精度は低い．NP-T モデルでは，$2^{19} \le N \le 2^{23}$ で誤差は最大 7%に収まる．FFTE では内挿範囲では N-T モデルは実用的な精度である．P-T モデルは N に関わらず精度が低い．NP-T モデルは内挿，外挿共に精度が高く実用的と言える．

HPL の評価評価を図 5 に示す．N-T モデルは $1600 \le N \le 8000$ で最適構成を予測できているが，$N = 9600$ では誤差が 100%を超えている．P-T モデルは $N = 2400$ で誤差が 26%，それを除けば誤差は最大 12%に収まっている．NP-T モデルは $1600 \le N \le 9600$ で誤差は最大 12%に収まっている．N-T モデルの外挿を除けば，全モデルとも実用的である．基本的には岸本らの結果[7] が再現されている

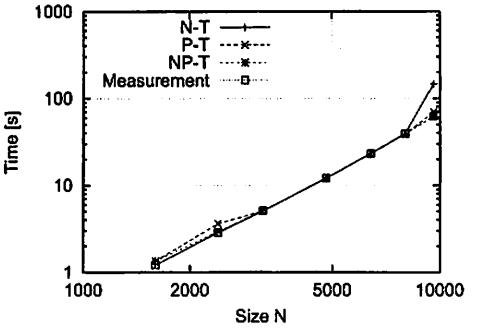


図 5 HPL

が，NP-T が P-T と同程度に正確であることは確認できた．

以上から，今回検討した範囲では NP-T モデルが適用範囲・精度の両面で優れていることがわかった．

5. おわりに

本研究で新たに提案した NP-T モデルは，先行研究[7][8] で提案された P-T モデルより正確に実行時間をモデル化することができ，実際に本研究の評価結果で見る限りは P-T モデルと同等以上のクラスタ構成を予測することができた．また，NP-T モデルは P-T モデルと同じ測定データから構築できるため，構築に必要な測定時間が短い．NP-T モデルは P-T モデルより構築可能な測定条件が広いため，P-T モデルが構築できない測定条件においても構築できる場合がある．

本研究では 2 種類のプロセッサを含む比較的小規模な不均一クラスタで評価を行ったが，現在 3 種のプロセッサを含む不均一クラスタに関しても評価を行っている．この不均一クラスタにおいても NP-T モデルに優位性があるか，検証を行う予定である．さらに大規模な不均一クラスタにおける評価に関しては，今後の課題とする．

本研究では NP-T モデルを提案したが，モデルに関してはまだまだ改良の余地がある．より正確で応用性の広いモデルが構築できないか，検討する必要がある．また，現在はサブクラスタがある (等価な PE が複数台ある) ことを前提としてモデル構築を行っているが，より不均一なクラスタからモデル構築を行うことは今後の課題である．

本研究では，均一で広帯域なネットワークを仮定してモデル化を行ったが，不均一な構成のネットワーク上でのモデルや，ネットワークトポロジを含むモデル，帯域制限のある場合の実行時間のモデル化等は重要なテーマである．これらも今後の課題とする．

謝辞 本研究の一部は，21 世紀 COE プログラム “インテリジェントヒューマンセンシング”，および科学研究費補助金 (基盤 (C)(2)16500029) によるものである．

参考文献

1) Kalinov, A. and Klimov, S.: Optimal mapping of a parallel application processes onto heterogeneous platform, *Proc. 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS 2005)*, IEEE Computer Society (2005). (CD-ROM).
2) 笹生健，松岡聡，建部修見：ヘテロなクラスタ環境における並列 LINPACK の最適化，情報処理学会研究報告 2001-HPC-86, pp.49–54 (2001).
3) Petitet, A., Whaley, R.C., Dongarra, J. and Cleary, A.: HPL - A Portable Implementation of the High-Performance Linpack Benchmark for Distributed-Memory Computers, `http://www.netlib.org/benchmark/hpl/`.
4) 笹生健，松岡聡：ヘテロなクラスタ環境における NAS Parallel Benchmarks の適用化，情報処理学会研究報告 2003-HPC-93, pp.1–6 (2003).
5) 大滝雄介，高橋大介，朴泰祐，佐藤三久：ヘテロなクラスタ環境における Strassen の行列積アルゴリズムの並列化．情報処理学会論文誌コンピューティングシステム，Vol.45, No.SIG 6 (ACS 6), pp.122–133 (2004).
6) Cuenca, J., Gimenez, D. and Martinez, J.-P.: Heuristics for Work Distribution of a Homogeneous Parallel Dynamic Programming Scheme on Heterogeneous Systems, *Parallel Computing*, Vol.31, pp.711–735 (2005).
7) Kishimoto, Y. and Ichikawa, S.: Optimizing the Configuration of a Heterogeneous Cluster with Multiprocessing and Execution-Time Estimation, *Parallel Computing*, Vol.31, No.7, pp. 691–710 (2005).
8) Kishimoto, Y. and Ichikawa, S.: An Execution-Time Estimation Model for Heterogeneous Clusters, *Proc. 18th Int'l Parallel and Distributed Processing Symposium (IPDPS 2004)* (2004). (CDROM).
9) Gustafson, J.L. and Todi, R.: Conventional Benchmarks as a Sample of the Performance Spectrum, *The Journal of Supercomputing*, Vol.13, No.3, pp.321–342 (1999).
10) 姫野龍太郎：Himeno BMT, `http://accc.riken.jp/HPC/HimenoBMT/index.html`.
11) 高度情報科学技術研究機構 (RIST)：HPC-MW 検証ツール hpcmw-solver-test, `http://hpcmw.tokyo.rist.or.jp/index.html`.
12) Takahashi, D.: FFTE: A Fast Fourier Transform Package. http://www.ffte.jp/.