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

2007－ARC－172 (14)
2007－HPC－109 (14)
2007／3／1

実行時間予測モデルの構築法の改善

河合 裕† 市川周一†

既存の並列応用の多くは均一並列環境を前提としているため，不均一クラスタ上で実行すると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起動することで負荷の均衡化が望めるが，最適に負荷を分散することは難しい．高橋と市川は実行時間予測モデルを構築し，4つの科学技術応用 (HPL, CFD, FEM, FFT) について (準) 最適構成を予測できることを示した．しかしモデル構築のために均一なサブクラスタを必要とする上，大きく精度が低下する場合があった．本研究では，不均一クラスタ自体からモデルを構築する方法と，精度低下を防ぐモデル構築手法を検討する．評価の結果，従来と同等以上のモデルを構築し，より高い精度で (準) 最適構成を予測することができた．

Improvements of Execution-time Estimation Model Construction

Yuu Kawai † and Shuichi Ichikawa†

The performance of a parallel application for homogeneous environment is degraded by load imbalance on heterogeneous clusters. Such load imbalance can be alleviated by invoking multiple processes on fast PEs in a heterogeneous cluster. Takahashi and Ichikawa constructed execution-time estimation models for four scientific applications, with which the sub-optimal configurations of heterogeneous clusters were estimated. However, their models have to be extracted from homogeneous subclusters, and the precisions of models are very sensitive to the fluctuations in measurement results. This study examines a method to construct models from heterogeneous cluster itself, and a method to construct more robust models. The derived models were shown to be superior to the previous models.

1. はじめに

演算性能や通信性能が異なる要素プロセッサ (PE) で構成されるクラスタを**不均一クラスタ**と呼ぶ．不均一クラスタは，余っているPCを集めてクラスタを構築したり，既存クラスタに高性能PEを追加する場合など，多くの局面で利用可能である．しかし，既存の並列応用の多くは均一なPEを前提にしており，各PEに負荷を均等に分散するため，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性能が低下する．不均一クラスタ上の性能を改善するには，PEの演算性能に応じて負荷を不均一に分散する必要がある．

不均一な負荷分散には大きく分けて二つの方法がある．ひとつは1プロセスに割り当てる問題領域の大きさを変更する方法である．しかし，この方法は応用プログラムの書き換えが必要になるうえ，書き換えに伴うデバグや性能チューニングなどに大きな手間がかかる．もうひとつは，高速なPEに複数のプロセスを起動することで負荷分散を行う方法 (マルチプロセス法) である．この方法では応用プログラムを書き換える必要がなく，1台のPEに起動するプロセス数を変更するだけで簡単に実現することができる．また，チップレベル並列性 (マルチスレッドやマルチコア) の利用にも適していると考えられる．

マルチプロセス法では，どのPEにいくつプロセスを起動するかが問題となる．岸本ら[1)2)]は，High-Performance Linpack Benchmark[4)] (HPL) の実行時間から実行時間予測モデルを構築し，不均一クラスタ上で最適なプロセス構成を予測できることを示した．高橋ら[3)]は，岸本らの手法をHPL以外の科学技術応用で検証するとともに，岸本らのモデルを改良して予測精度を改善した．

しかし，これまでの手法では均一PEを使用して予測モデルを構築していたため，利用可能な状況が限られていた．また，モデル構築に必要な実測値が少ない場合や，実測値に誤差が含まれる場合に，予測精度が著しく低下する場合があった．本研究では，不均一クラスタから直接モデルを構築する方法を提案し，更に予測精度の低下を回避する手法を提案する．

† 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

2. 実行時間予測モデル

2.1 高橋らの実行時間予測モデル

応用の問題サイズを N とする．不均一クラスタ中の等価な PE のグループ (サブクラスタ) を G_i であらわし，G_i 中で実際に計算に使用する PE の台数を P_i とする $(0 \leq P_i \leq |G_i|)$．G_i 内の各 PE には同数のプロセスを起動するものとし，この数を M_i で表す．このとき不均一クラスタ内の総プロセス数 P は，$P = \sum_i P_i M_i$ で与えられる．不均一クラスタ全体の実行時間を T としたとき，ある N について T を最小化する構成 (P_i, M_i) を予測することが研究の目的である．サブクラスタ G_i の実行時間 T_i を，N，P，M_i の関数で近似することができれば，全体の実行時間 T は $\max_i T_i$ で見積もることができる．T の近似式を実行時間予測モデルと呼ぶ．全ての可能な構成に対して予測モデルを構築できれば，構成 (P_i, M_i) と問題サイズ N について，T を最小化する構成を予測することができる．

以下，HPL を例にとる．HPL の実行時間は以下の式で見積もることができる[1)2)]．

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

式 (1) において，ある M_i の実行時間 T_i を N と P の関数で表すと次の式になる．

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

高橋らは式 (2) を NP-T モデルと名づけた．式 (2) には k_0 から k_9 までの 10 個の定数項があるが，これらの係数は実測値を用いて最小二乗法で決定する．NP-T モデルでは G_i および M_i ごとに NP-T モデルを構築する．PE が 1 台 $(P = \exists M_i)$ の場合は通信時間がなくなることを考慮して，定数項の決定には均一 PE 2 台以上の実測値を使用する．

2.2 既知の問題点

式 (1) は N の 3 次式であるため，0 次の項を含め N について最低でも 4 つ，P についても同様に最低でも 3 つの実測値が必要になる．通常 N は容易に変更できるため大きな問題ない．しかし，P の異なる実測値を得るには，PE の台数を変化させる必要があるため最低でも 4 台，応用の制限によってはそれ以上の台数の均一 PE が必要となる．

また，実測値の数が足りていたとしても，その数が少ない場合や実測値に誤差がある場合にはパラメータが正しく抽出されない．パラメータの抽出に失敗した典型的な例を**図 1** に示す．これは $P = 8$ としたときの Himeno BMT[5)] の NP-T モデルである．図中の

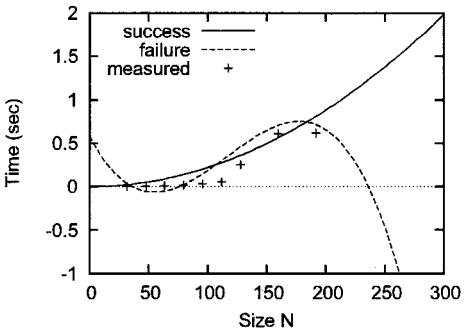

図 1 モデル破綻の例

success はパラメータの抽出に成功した場合，failure は失敗した場合を示す．成功例では問題サイズ N の増加にともない予測実行時間が増加しているのに対し，失敗例では問題サイズが大きくなると予測実行時間が負の値になっている．本研究では，図 1 のようにモデルの高次の項が負の値になり，負の実行時間を予測することをモデルの破綻と呼ぶ．モデルの破綻が生じるとその構成の予測実行時間が最小となり，この構成が最適構成として予測されてしまうため，真の最適構成が予測できない．

3. 改善方法

本研究では NP-T モデルの式を変更せず，モデル構築の方法を改善することを検討する．NP-T モデルはネットワークトポロジやキャッシュの構造を考慮していないため，それらを考慮したモデルを使うことで予測精度が改善される可能性はあるが，それについては今後の課題とする．

3.1 不均一クラスタからのモデル構築

高橋らの手法では，NP-T モデルを構築するために等価な PE 4 台以上からなる均一サブクラスタが必要であった．本研究では，低速な PE を含む不均一クラスタの実行時間が，同数の低速 PE で構成される均一サブクラスタの実行時間と (ほぼ) 同じであると仮定する．この仮定は，「不均一クラスタ内の各 PE に同数のプロセスを実行すれば，最も遅い PE の実行時間がサブクラスタの実行時間として観測される」と言い換えてもよい．この仮定が正しければ，不均一サブクラスタの実行時間からモデルを構築できることになる．

この手法では，高速な PE を利用して低速な PE のモデルを作成することができる．副次的効果として，利用可能な高速 PE をすべてを使うことでパラメータ抽出に用いるデータ点数を増やして，低速 PE のモデル精度が向上することも考えられる．

しかし，不均一サブクラスタでは均一サブクラスタと通信のタイミングが異なるので，上記の仮定が成り

表 1 NP-T モデル[3)]

ベンチマーク	モデル式
HPL	$T_i(N,P)\|_{M_i} = \frac{1}{P}\cdot(k_0N^3+k_1N^2+k_2N+k_3)+P\cdot(k_4N^2+k_5N+k_6)+k_7N^2+k_8N+k_9$
Himeno BMT	$T_i(N,P)\|_{M_i} = \frac{1}{P}\cdot(k_0N^3+k_1N^2+k_2N+k_3)+k_4N^2+k_5N+k_6+k_7\log P$
hpcmw-solver-test	$T_i(N,P)\|_{M_i} = \frac{1}{P}\cdot(k_0N^3+k_1N^2+k_2N+k_3)+k_4N^2+k_5N+k_6+k_7\log P$
FFTE	$T_i(N,P)\|_{M_i} = \frac{1}{P}\cdot(k_0N\log N+k_1N+k_2)+k_3P+k_4N+k_5N^{\frac{1}{3}}+k_6$

表 2 評価環境

	サブクラスタ G_1	サブクラスタ G_2	サブクラスタ G_3
CPU	Pentium 4 3.6GHz	Xeon 2.8GHz	Celeron M 1.5GHz
台数	8	8	8
メモリ	1GB	1GB	1GB
NIC	1000BASE-T	1000BASE-T	1000BASE-T
OS	Fedora Core 4	Red Hat Linux 9	Fedora Core 5
開発環境	Intel C/C++ Compiler 9.0, Intel Fortran Compiler 9.0, mpich 1.2.7p1		
P_i, M_i の範囲	$0\le P_1\le 2,\ 0\le M_1\le 3$	$0\le P_2\le 4,\ 0\le M_2\le 2$	$0\le P_3\le 2,\ 0\le M_3\le 1$

表 3 問題サイズ

ベンチマーク	モデル構築時	評価時
HPL	400〜6400 (9 点)	400〜9600 (11 点)
Himeno BMT	32〜192 (9 点)	32〜256 (11 点)
hpcmw-solver-test	60〜442 (7 点)	60〜600 (12 点)
FFTE	2^{12}〜2^{20} (9 点)	2^{12}〜2^{23} (12 点)

立つかどうか自明ではない．実行時間の差がパラメータ抽出に影響し，予測精度が低下する可能性もある．均一クラスタから構築したモデルと同様の精度が得られるかどうか実応用で評価する必要がある．

3.2 非負最小二乗法によるパラメータ抽出

予測時間が負の値にならないようにするためには，モデルの各パラメータ $k_0, k_1, \ldots \ge 0$ という条件をつければよい．これは，計算時間や通信時間が単調増加すると考えれば，妥当な制限である．そこで本研究では，抽出されるパラメータに非負制約を付けた非負最小二乗法[8)]を用いることを試みる．非負最小二乗法を使用してモデルを構築すると，負の実行時間が予測されなくなり，極端な精度低下を防ぐことが期待できる．

一方，非負最小二乗法でパラメータを抽出すると，通常の最小二乗法で抽出したモデルよりも一般に残差が大きくなる．そのため，大きなモデル破綻が防げる代わり，予測精度が低下する可能性がある．これについても実際に評価して優劣を比較する必要がある．

4. 評価方法

本研究のモデル構築には，応用の問題サイズ N と総プロセス数 P が変更可能である必要がある．今回の実験では，上記の条件を満たす次の 4 つのベンチマークを評価対象とした．

HPL[4)] 分散メモリ並列計算機用の Linpack ベンチマークで，倍精度浮動小数点演算で線形代数方程式を解く性能を測定する．

Himeno BMT[5)] 非圧縮流体解析コードのベンチマークで，Poisson 方程式を Jacobi 反復法で解く性能を測定する．

hpcmw-solver-test[6)] 3 次元弾性解析を評価するベンチマークで，有限要素法で問題を解く性能を測定する．

FFTE[7)] $2^p3^q5^r$ 要素の複素離散フーリエ変換を行うライブラリで，実験では同梱されている 1 次元 DFT のテストプログラムを用いる．FFTE では P は 2 の冪乗でなければならないので，その条件を満たす構成だけを評価する．

これらは高橋らによって評価されたもので，今回の実験では追試する意味含め，評価対象とした．高橋ら[3)]によって求められた NP-T モデル式を**表 1** に示す．

評価に使用した不均一クラスタの構成を**表 2** に示す．今回の実験では計算時間への影響を避けるため，Xeon および Pentium 4 の Hyper Threading は使用しない．また，ネットワークトポロジによる問題の複雑化を避けるため，全ての PE を一台のスイッチに接続している．モデル構築のための測定は，**表 3** に示す問題サイズで行った．評価はモデル構築時の測定に 2 点から 5 点を追加し，外挿についても行う．

Celeron M のモデルは Celeron M (2 台) と Xeon (4 台)，Pentium 4 (2 台) の合計 8 台を使って構築し，Xeon のモデルは Xeon (4 台) と Pentium 4 (4 台) の合計 8 台を使って構築する．最も高速な Pentium 4 のモデルは，Pentium 4 (8 台) の均一サブクラスタから構築する．これらのモデルと，8 台の PE からなる均一サブクラスタを用いて構築したモデルの比較を行う．

パラメータ抽出に使用する非負最小二乗法は文献 8) で紹介されているものを利用する．先行研究で用いられている GSL[9)] の `gsl_multifit_linear()` による最小二乗法と比較し評価を行う．

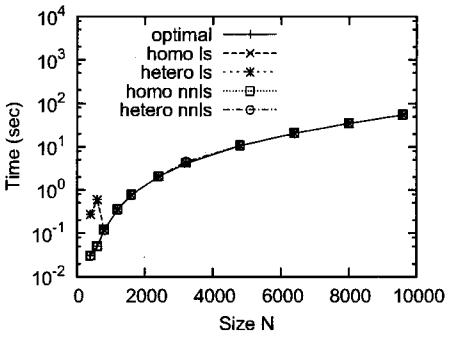

図 2 HPL の評価結果

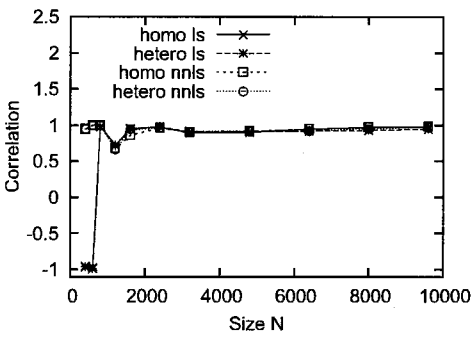

図 3 HPL の相関係数

5. 評価結果

最適構成の探索は組合せ最適化問題の一種であり，一般には計算困難である．しかし今回の実験で用いる不均一クラスタでは組合せ総数が 188 通りと少ないため，特別な探索アルゴリズムは利用せず，全構成の予測時間を求めて最適構成を予測する．探索に要する時間は，評価対象の応用に対していずれも 0.1 秒以下と十分に小さい．

HPL 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**図 2** に示す．図中の “optimal” は実測最適構成の実測時間を示す．均一サブクラスタからモデル構築する場合を “homo”，不均一クラスタ自体からモデル構築する場合を “hetero” で表す．通常の最小二乗法でパラメータを抽出する場合を “ls”，非負最小二乗法の場合を “nnls” で表す．高橋らの手法は均一サブクラスタの実測値を使い，通常の最小二乗法でパラメータを抽出するので “homo ls” に該当する．

いずれの手法も $N \geq 800$ では (準) 最適構成が予測されている．nnls は評価した全域にわたり良好な精度を示している．ls では $400 \leq N \leq 600$ で誤差が 100%を超え予測に失敗しているが，実行時間は 1 秒未満であり実行時間差は小さい．基本的には高橋らの結果が再現されており精度はよい．

予測時間と実測時間の相関係数を**図 3** に示す．図 3 では，モデルの予測実行時間を τ，最適構成の実測時間を T_{opt} としたとき $\tau \leq 2T_{opt}$ となる構成 (準最適構成) についてだけ相関係数を求めている．実行時間が大きく最適構成から掛け離れた構成まで含めると，相関係数が下がって手法の精度を正しく評価できないためである．$N \leq 800$ では上記の条件を満たす構成が 3 点しかないため，ここでは 10 点以上の構成を含む $N \geq 1200$ に注目する．いずれの手法も $N \geq 1600$ で 0.9 を超える高い相関を示している．HPL についてはいずれの手法も精度が高く，homo と hetero は同程度の精度をもつといえる．

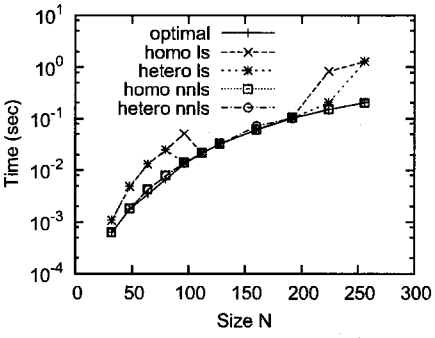

図 4 Himeno BMT の評価結果

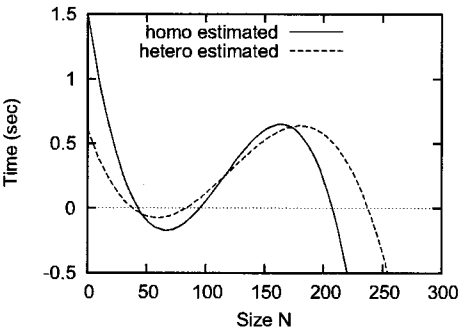

図 5 ls で破綻した HimenoBMT のモデル

Himeno BMT の評価結果を**図 4** に示す．ls は homo, hetero ともに $112 \leq N \leq 192$ で最大誤差 5%と準最適構成が予測されているが，外挿範囲である $N = 256$ では誤差 500%と悪くなっている．ls ではいずれもモデル破綻が生じ，外挿範囲の予測に失敗している．破綻したモデルを**図 5** に示す．これは，$P = 4$ としたときの Xeon $M_i = 2$ のモデルである．このモデルが $N \geq 200$ で負の実行時間を予測するため，誤って最適構成として予測されてしまい，結果として図 4 の

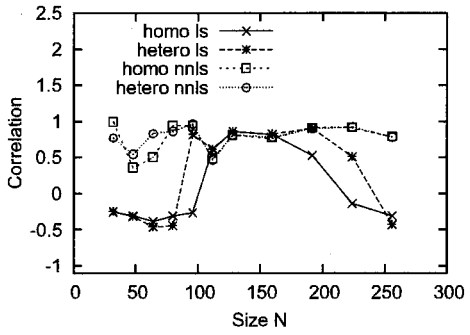

図 6 Himeno BMT の相関係数

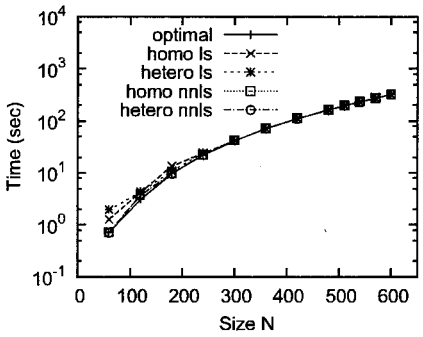

図 7 hpcmw-solver-test の評価結果

ように大きな予測誤差を示すことになった．一方 nnls では予測実行時間が負になることはないため，homo, hetero ともに $192 \leq N \leq 256$ で最適構成の予測に成功している．

予測時間と実測時間の相関係数を**図 6** に示す．図 6 では $\tau \leq 2T_{opt}$ の準最適構成について相関係数を求めているが，Himeno BMT では $N \leq 64$ で条件を満たす構成が 5 点しかないため，精密な議論は難しい．10 点以上の構成を含む $N \geq 80$ の場合，ls では外挿範囲の $N \geq 224$ で相関が低くなっているが，nnls では 0.8 と高い相関を維持している．Himeno BMT については nnls ではモデル破綻がなくなり，homo でも hetero でも同程度に高い予測精度を得た．

hpcmw-solver-test の評価結果を**図 7** に示す．hpcmw-solver-test は本来 $N \times N \times N$ の問題を扱うが，本研究では実行時間，記憶容量の都合から $N \times N \times 1$ の問題を扱う[3]．ls では $240 \leq N \leq 600$ で homo, hetero ともに誤差は最大で 8%と (準) 最適構成が予測できている．nnls では $180 \leq N \leq 600$ で homo, hetero ともに最適構成が予測されている．

$\tau \leq 2T_{opt}$ の準最適構成の予測時間と実測時間の相関係数を**図 8** に示す．hpcmw-solver-test では $N \geq 60$ で 10 点以上の構成を含んでいる．いずれの手法も $300 \leq N \leq 600$ で 0.8 を超える高い相関を示しており，精度の高さが確認できる．hpcmw-solver-test では，HPL 同様いずれの手法も高い精度を示し，homo と hetero で同程度の精度があるといえる．

FFTE の評価結果を**図 9** に示す．ls は homo, hetero ともに $2^{17} \leq N \leq 2^{20}$ で最大誤差 1%と準最適構成が予測されているが，外挿範囲の $2^{21} \leq N \leq 2^{23}$ では誤差が 80%を超えている．FFTE では HimenoBMT と同様にモデル破綻が生じているため，外挿範囲で誤差が大きくなっている．nnls では $2^{12} \leq N \leq 2^{23}$ で最大誤差 1%と大きく改善されており，全ての N で homo, hetero ともに同じ構成が予測されている．

予測時間と実測時間の相関係数を**図 10** に示す．FFTE では，準最適構成の条件を $\tau \leq 2T_{opt}$ とする

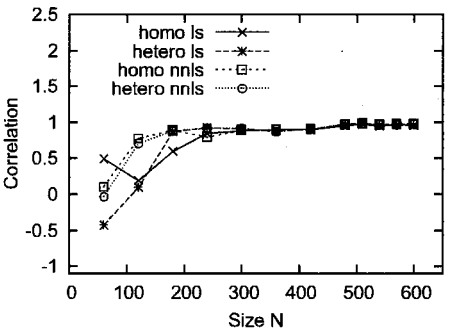

図 8 hpcmw-solver-test の相関係数

と，$2^{12} \leq N \leq 2^{20}$ で条件を満たす構成が 4 点しかないため，FFTE に限り条件を緩和して $\tau \leq 3T_{opt}$ を準最適構成とした．この条件でも $N \leq 2^{15}$ では条件を満たす構成が 3 点しかないが，$N \geq 2^{16}$ では 10 点以上の構成が確保できている．ls では homo, hetero ともに $2^{17} \leq N \leq 2^{20}$ で 0.8 と高い相関係数を示しているが，外挿範囲の $2^{21} \leq N \leq 2^{23}$ では 0.2 以下となっており精度が悪い．nnls では，hetero は $2^{22} \leq N \leq 2^{23}$ で 0.8 を超える高い相関を示している一方，homo は $N = 2^{23}$ で 0 近くまで相関が悪くなっている．このときの予測モデルを**図 11** に示す．これは $P = 8$ としたときの Xeon $M_i = 2$ のモデルである．$N = 2^{17}$ 付近から homo と hetero で予測時間がずれ始め，$N = 2^{23}$ では homo の予測時間は hetero の 76%になっている．このことから homo ではいくつかの構成で実行時間を小さく見積もり，その結果相関係数が下がったと考えられる．FFTE については，nnls でモデル破綻がなくなり ls に比べて精度が向上したが，精度面では依然として課題が残る結果となった．

6. おわりに

本研究では，不均一クラスタからモデル構築する手

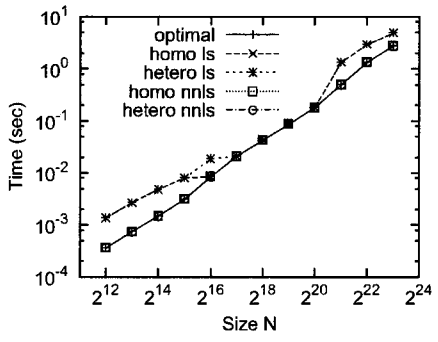

図 9 FFTE の評価結果

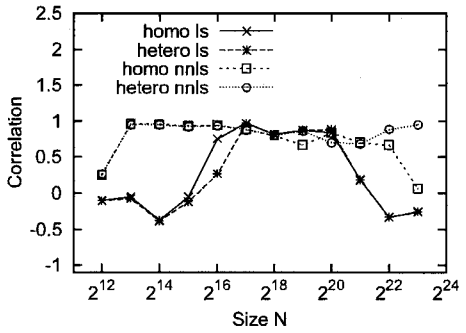

図 10 FFTE の相関係数

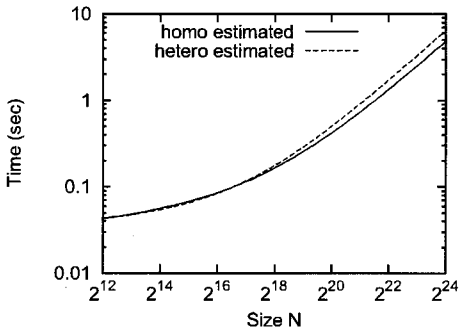

図 11 nnls での FFTE のモデル

法を提案し，モデルの破綻がないケースについては均一クラスタからモデルを構築した場合と同様の精度が得られることを示した．この手法は不均一クラスタにおける性能のボトルネックを利用するため，最も高速な PE に対しては適用できない．高速な PE のモデルの構築が今後の課題である．本研究では非負最小二乗法を用いることでモデルの破綻を抑制し，予測精度の向上を試みた．従来手法では予測に失敗していた Himeno BMT，FFTE についても (準) 最適構成の予測に成功し，有効性が確認できた．

NP-T モデルは均一なネットワークを仮定しているため，本研究では全ての PE が 1 台のスイッチに接続され，NIC が 1000BASE-T に統一されている均一なネットワーク環境下で評価を行った．しかし，PE の台数が大きくなると 1 台のスイッチに全ての PE を接続することは困難であるため，スイッチが多段になる場合などの通信不均一な環境を考慮する必要がある．本研究ではモデル自体の変更は行わず，構築方法を変えることで不均一クラスタからのモデル構築と予測精度の向上を図ったが，通信をモデル化するにはこのような方法だけでは対応は難しい．通信を考慮した新たなモデル構築が今後の課題である．

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・基盤研究 (C)(2)16500029 の援助により行われた．

参 考 文 献

1) 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).
2) 岸本芳典，市川周一: 不均一クラスタ上での実行時間予測モデルとその改良，情報処理学会研究報告 2004-HPC-97, pp.73–78 (2004).
3) 高橋翔，市川周一: 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，情報処理学会研究報告 2006-HPC-105, pp.97–102 (2006).
4) Petitet, A., Whaley, R.C., Dongarra, J. and Cleary, A.: HPL – A Portable Implementation of the High-Performance Linpack Benchmark for Ddistributed-Memory Computers, http://www.netlib.org/benchmark/hpl/.
5) 姫野龍太郎: Himeno BMT, http://accc.riken.jp/HPC/HimenoBMT/.
6) 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/.
7) Takahashi, D.: FFTE: A Fast Fourier Transform Package, http://www.ffte.jp/.
8) Lawson, C.L. and Hanson, R.J.: Solving Least Squares Problems (Prentice-Hall Series in Automatic Computation), Prentice Hall (1974).
9) Free Software Foundation: GSL – GNU scientific library, http://www.gnu.org/software/gsl/.