Vol.2016-HPC-155 No.19
2016/8/9

モデルパラメータに非負制約を課した回帰モデルによる大規模並列計算の性能予測

折居 茂夫[†1]　今村 俊幸[†1]　山本 義郎[†2]

本研究の目標は大規模並列計算の性能予測方法を確立することである．これまで，予測モデル構築方法として，過剰適合を抑制して予測精度を向上する「非負制約を課したモデル化方法を提案した[1]．この方法をスパースモデリングに使われる lasso を使ったモデル化と比較すると，予測精度が向上することを示した[2]．本講演では大規模並列計算として HPL を SGI ICE X で実行した場合と EigenExa [3]を「京」で実行した場合を用い，上記の提案モデル化方法を適用して性能予測を試みた結果を報告する．

キーワード：大規模並列計算，性能予測，モデル化方法，lasso，過剰適合，非負制約

Performance Prediction of Large-Scale Parallel Computing by Regression Model with Non-Negative Model Parameters

SHIGEO ORII[†1]　TOSHIYUKI IMAMURA[†1]
YAMAMOTO YOSHIRO[†2]

The purpose of this study is to establish a methodology for the performance prediction model for large-scale parallel computing. We have proposed a method that improves predictive ability by suppressing overfitting with non-negative constraint [1]. We have compared the proposed method with the lasso, which is often used in the sparse modeling field, and found that the predictions of the proposed method accurate better than the lasso's ones [2]. Using the proposed method, we make predictive models of HPL running on an SGI ICE X and EgenExa [3] running on the "K" computer as the modeling examples of large-scale parallel computing. The results of that are reported in this paper.

Keywords: Large-Scale Parallel Computing, Performance Prediction, Modeling, lasso, Overfitting, non-negative constraint

1. はじめに

並列処理による計算時間の短縮は，プロセッサやコア数の増加と供に減少する時間と，並列処理で意図した計算結果を得るために加えられた通信や同期等による処理時間即ち，並列オーバーヘッドのせめぎ合いにより決まる．この結果，計算問題を並列処理で解こうとしたとき，ある並列計算機に対し，複数の並列処理の計算法（アルゴリズム）が存在する．その中でどのアルゴリズムが効果的か，最短処理時間となるかということが重要となる．また逆に，ある計算問題とそれを解くアルゴリズムがあるとき，どのような並列計算機だったら最短処理時間が得られるかということが重要となる．そこで並列処理における計算時間を予測することが重要となり古くから研究されてきた．その方法の一つに，モデル関数を用い，小規模の問題を解いたときの処理時間を基にしてモデルパラメータを決定して大規模の計算問題を解いたときの時間を予測する方法があり，たくさんの方法が提案されてきた．

一方提案されたこれらの予測方法は，提案した論文の予測事例はあるが，事例以外の並列計算で適用され有効であったと報告された事例を筆者らは知らない．その最大の原因の 1 つが，モデルパラメータを決定するときに生じる「過剰適合」にある．「過剰適合」はモデルを構築するために用いた観測データに含まれるノイズが，モデル中の冗長な基底関数でモデル化されることにより生じる．表 1，図 1 は過剰適合の典型的な例である[1]．モデル化のための観測データとして，真のモデル $y^0= x^2$ を基に生成したシミュレーションデータ **y** について，多項式関数 $f(x)=a_0+a_1x+a_2x^2+a_3x^3+a_4x^4+a_5x^5$ (M=5)でモデル化する．**y** の要素は，1.1 から 3.1 を等間隔に 8 点(N=8)で分割した x_i に対して $y_i = x_i^2 + d_i$ (i=1,…,8)で，平均 0，標準偏差 σ =0.01 の正規分布に従う揺らぎ d_i を与えて生成した．表 1 はこの **y** と $f(x)$を用いて最小二乗法で得られたモデルパラメータである．このモデルは真のモデル $y^0=x^2$ とは大きく異なる．

表 1　LSM によるモデルパラメータ（σ=0.01）

a_0	a_1	a_2	a_3	a_4	a_5
4.30	−11.2	12.2	−5.42	1.29	−0.120

図 1 は観測データの揺らぎがモデル化されるため，×で

†1 理化学研究所計算科学研究機構
RIKEN Advanced Institute for Computational Science
†2 東海大学
Tokai University

表した観測データの内挿には精度良いモデルとなるが，外挿領域では x^2 から大きく外れ出し，やがて予測値は負となってしまう．

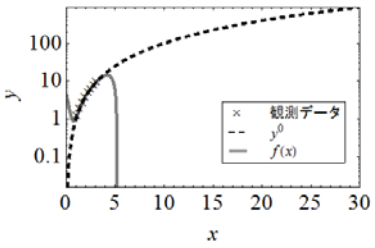


図 1　「過剰適合」の例

そこで筆者らは，モデルパラメータに非負制約を課すことにより「過剰適合」を抑制する方法 LP を提案した[1]. LP を適用するとモデルパラメータは表 2 となる．

表 2　LP によるモデルパラメータ（σ=0.01）

a_0	a_1	a_2	a_3	a_4	a_5
0	0	0.994	0.00212	0	0

幸い我々は時間を予測するため，モデル関数は非負である．そこでモデル関数を構成する基底関数を工夫することにより，全基底関数を非負とし，そのモデルパラメータに非負制約を課すことができる．結果，観測データの 100 倍のレンジの予測が視野に入ってきた[1].

LP のもう一つの特徴は，「変数選択」ができることにある．最小二乗法を用いた場合，モデル化に用いた全ての基底関数が残差平方和を最少にするために使われる．その結果全てのモデルパラメータが値を持つ．一方 LP ではモデル化には不必要な基底関数のモデルパラメータ値は整数の 0 値となる．従って 0 値の項を削除する「変数選択」が可能となる．表 2 の 0 値がこれに当たる．これによりモデル関数を構築する基底関数に複数の候補を混在して取捨選択することが可能となる．このような「変数選択」は L1 正則化で実現できることが知られている．そこで我々は L1 正則化を具現化する lasso[4]と提案方法を比較し，モデル化のための観測データが変わった場合に LP の予測の方が lasso より良く収束することを示した[2]. 図 2-1 は HPL(High performance Linpack) [5]を PRIMERGY RX200S5 で実行した場合の予測である．×は確認のため測定したデータで実線は予測データである．モデル化のための観測データは任意の 8 点を用いた 5 ケースのモデルである．lasso と LP の予測を比べると，LP の予測がより収束していることがわかる．図 2-2 は HPL を PRIMEPOWER HPC2500 で実行した場合の予測である．この図は，並列オーバーヘッドが大きくなり下に凸の現象が生じた場合の予測となるが，これに対しても LP の予測が lasso より収束していることを示す．

このように予測モデル構築に必要なことは，(1)モデルを外挿して予測ができるようにするための変数選択（モデルパラメータの選択），(2)得られたモデルパラメータ値が計算誤差等に左右されないものであることであり，これらは LP によりクリアできるものと考える．

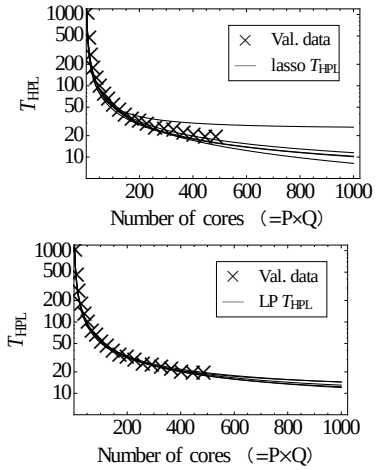


図 2-1 HPL と PRIMERGY RX200S5 における予測.

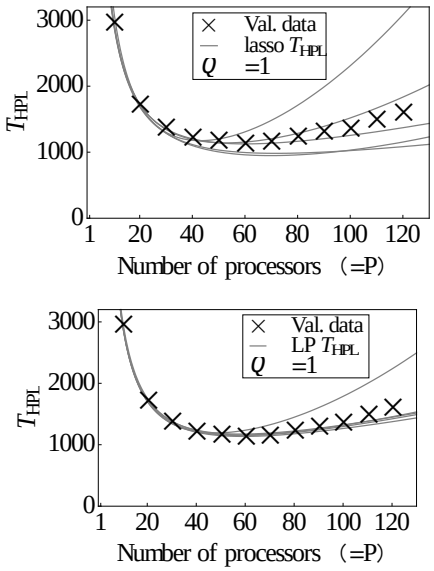


図 2-2 HPL と PRIMEPOWER HPC2500 における予測.

そこで(3)モデル関数が現象を記述できることが重要となる．特に，プログラムと計算機の組み合わせにより複雑になることが予想され，モデル関数を構成する未知の基底関数が必要になると考えられる．実際本報告で用いた事例においても今までの経験では捉えきれない現象が生じ，モデル関数に基底関数を追加した．本論文ではこの関数を発見したので，その方法を報告する．大規模並列計算の性能予測の可能性を探るため，HPL に加え，最新のアルゴリズムを実装したアプリケーションプログラム EigenExa [3]を用いることとし，EigenExa を理研の「京」で，HPL を統計数理研究所の「SGI ICE X」で実行したデータを用いること

にした.

2 章で LP を簡単に紹介する．3 章では LP を用いた予測モデル構築方法を示す. 4 章で事例を示す. 5 章で議論とまとめを行う.

2. 非負制約によるモデルパラメータ決定方法

回帰モデルを $y(x)=a_0+\Sigma a_k f_k(x)$ $(k=1,\ldots,M)$とする．ここに y は M 個の基底関数と説明変数 x の目的変数である．係数 a_k は，観測データ (x_i, y_i) $(i=1,\ldots,N)$に対するモデルパラメータである.

このモデルのモデルパラメータ決定において生じる過剰適合を抑制するため，我々はモデルパラメータに非負制約を課して決定することを提案したが，従来の最小二乗法ではしばしば誤った値を得た．そこでシンプレックス法を用いて残差の絶対値$|e_i|$ $(=|y_i-f(x_i)|)$の最大値を最小にする方法を提案した[1].

ここで $\mathbf{y}=[y_1, y_2, \ldots, y_N]^T$, $\mathbf{a}=[a_0, a_1,\ldots, a_M]^T$, $\mathbf{e}=[e_1, e_2, \ldots, e_N]^T$ とし，行列 $\mathbf{f}=\begin{bmatrix} 1 & 1 & \cdots & 1 \\ f_1(x_1) & f_1(x_2) & \cdots & f_1(x_N) \\ \vdots & \vdots & & \vdots \\ f_M(x_1) & f_M(x_2) & \cdots & f_M(x_N) \end{bmatrix}$ と表記し, $f_0(x_i)=1$ とすると, 残差を $\mathbf{e}=\mathbf{y}-\mathbf{f}^T\mathbf{a}$ と書くことができる.

ここで最大残差ε $(=\mathrm{Max}\ e_i)$とし，N 要素のベクトルを $\boldsymbol{\varepsilon}=[\varepsilon,\ldots,\varepsilon]^T$ とすると，$\boldsymbol{\varepsilon}$と $\mathbf{e}$ の関係は，連立残差不等式(1)と書ける.

$$\boldsymbol{\varepsilon} \geq |\mathbf{y}-\mathbf{f}^T\mathbf{a}| \quad (\mathbf{a}\geq 0) \tag{1}$$

モデルパラメータ $\mathbf{a}$ は$\boldsymbol{\varepsilon}$を最小にすることによって$\boldsymbol{\varepsilon}_{\min} \geq |\mathbf{y}-\mathbf{f}^T\mathbf{a}|$ $(\mathbf{a}\geq 0)$のように決定できる．ここに$\boldsymbol{\varepsilon}_{\min}=[\varepsilon_{\min},\ldots,\varepsilon_{\min}]^T$ である.

ここで式(1)の$\boldsymbol{\varepsilon}$を最小にするために線形計画法を用いることにし，式(1)を$-\boldsymbol{\varepsilon} \leq \mathbf{y}-\mathbf{F}^T\mathbf{a} \leq \boldsymbol{\varepsilon}$ののように線形化する．ここに $\mathbf{F}$ は $\mathbf{f}$ に x_i を代入した値の行列である.

式(2)は非負条件 $\mathbf{a}\geq 0$ を課した式(1)の$\boldsymbol{\varepsilon}$を，線形計画法のアルゴリズムの一つシンプレックス法を用いて最小にし，モデルパラメータを決定する式である．本報告書では以後これを LP と呼ぶことにする.

$$\begin{aligned} &\text{Min} \quad \varepsilon \\ &\text{s.t.} \quad \boldsymbol{\varepsilon} + \mathbf{F}^T\mathbf{a} \geq \mathbf{y} \\ &\qquad \boldsymbol{\varepsilon} - \mathbf{F}^T\mathbf{a} \geq -\mathbf{y} \\ &\qquad \mathbf{a}\geq 0 \end{aligned} \tag{2}$$

3. 予測モデル構築方法

これまで LP を用いた予測モデルのモデル関数は，計算アルゴリズムに基づく計算回数を式化したものであった．このモデル関数，構成する基底関数が冗長にあっても現象を記述できる基底関数が含まれていれば, LP の変数選択によりモデル関数を構成する基底関数を特定できる．このため方法 1 を行ってきた.

方法 1 ：複数の変数を持つモデル関数に LP を適用し，変数選択とモデルパラメータ決定を同時に実施

一方本事例で紹介する HPL を SGI ICE X で実行した場合と，EigenExa を京で実行した場合について，計算アルゴリズムに基づく計算回数に比例するとした関数だけではモデル化できなかった．そこで図 2 に示す方法 2 を考案した.

方法 2 : 1 変数関数となるよう変数を固定し，変数選択を行い，選択した基底関数を基に再度の変数選択とそのモデルパラメータの決定を行う．この手順は次の 4 ステップからなる.

1) プロセッサ数 p を固定して通信量を一定にし, 計算時間を問題の大きさ N に対してスケーリングし N に対する基底関数を発見する．（このモデルでは p に依存する処理時間は全て定数の切片としてモデル化される.）
2) N を固定してストロングスケーリングし，p に対する基底関数を発見する．（このモデルでは N に依存する処理時間は全て定数の切片としてモデル化される.）
3) 1)と 2)で発見した基底関数を組み合わせてモデル関数の候補を作成．例えば 1) の結果が $F_N(N)=c_1f_1(N)+c_2f_2(N)+c_0'$，2)の結果が $F_p(p)=c_3f_3(p)+c_4f_4(p)+c_0''$ となった場合，モデル関数の候補の候補は原則次式となる.

$$F_{Np}(N,p)=c_0+c_1f_1(N)+c_2f_2(N)+c_3f_3(p)+c_4f_4(p)+c_5f_1(N)f_3(p)+c_6f_1(N)f_4(p)+c_7f_2(N)f_2(N)+c_8f_2(N)f_4(p) \tag{1}$$

4) 3)のモデル関数候補の c_0~ c_8 を LP により決定し, モデルパラメータ決定と変数選択を行う.

方法 2 を図 3 に示す.

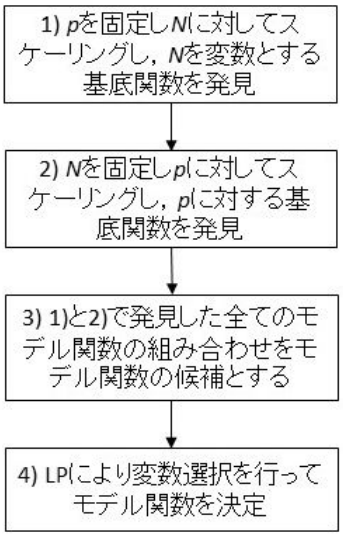


図 3 使用した予測モデル構築方法

4. 予測モデル構築事例

３章の予測モデル構築方法に至った 2 つの予測モデル構築事例を示す．２章の LP の機能検証に用いた計算機は，FUJITSU PROIMEPOWER HPC2500，FUJITSU PRIMERGY RX200S5 であった．これらの計算機で行った計算時間は，アルゴリズムから導かれる計算回数に比例し，通信時間は通信量に比例するとしてモデル化することができた．一方現在の大規模並列計算機はマルチコアを採用し，大規模化に伴い通信トポロジも複雑化している．これら現在の大規模並列計算を実現する並列計算機上で動き，それらの計算機に最適化されたプログラムの計算時間の予測モデルが構築できるかどうかを確かめるため, 4.1 節 HPL を SGI ICE X で実行した場合と，4.2 節 EigenExa を京で実行した場合についてモデル化を試みた．

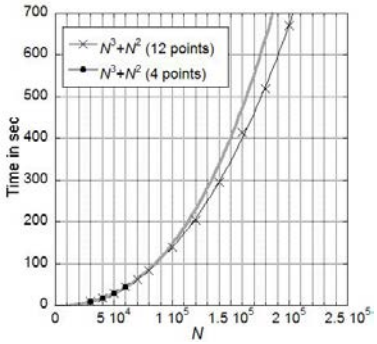
4.1 HPL を SGI ICE X で実行した場合

統計数理研究所の SGI ICE X の CPU は Intel Xeon E5-2697v2 で 12 コア 2.7GHz，1 ノードは 2CPU から成る．主メモリは 256GB である．ノード間の通信は 5 次元ハイパーキューブを 4×FDR InfiniBand×1 で実現している．使用した計算ノード 2 のノード数は 128 である．HPL は Intel c コンパイラでコンパイルし，Intel MKL の BLAS をリンクした．

図 3 のステップ 1 の結果の一部を図 4 に示す．HPL の計算時間は式(3) となり N^3+N^2 に比例する[5]. ここに γ, β_f, β_m はモデルパラメータである．

$$\gamma\frac{2N^3}{3PQ}+\beta_f\frac{N^2}{2P}+\beta_m\frac{3N^2}{2Q} \tag{3}$$

これを用いたモデル化が図 4 上である．×で示した観測データ 12 点全てを用いると細い実線のようによく一致するモデルを構築できる．一方この図は，●で示した 3 点の観測データを用いると，N が大きくなるに従い予測精度が落ちることを示す．そこで基底関数を追加してこれを改善することを試みた．試行錯誤の結果，$-N$ 項を導入すると予測精度が向上することを発見した．これを図 4 下に示す．



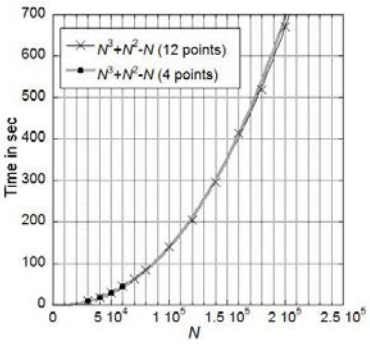


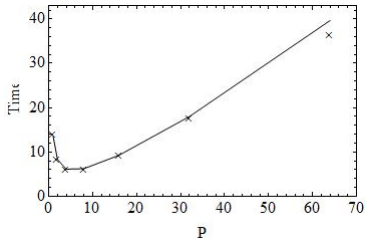
図 4 P=32, Q=2 の処理時間のモデル化　(図 3 ステップ 1)
（上：N^3+N^2 モデル，下：N^3+N^2-N モデル）

$-N$ 項は N が大きくなるに従い効かなくなる項である. 今までの HPC2500 と RX200S5 を使ったモデル化ではこの項は必要無かったので，この項が必要になるこの現象は SGI ICE X 由来すると考える．

図 3 のステップ 2 で用いた基底関数を式(4)に示す．モデルパラメータは a_f，a_m，b_f，b_m である．2~3 行目の Log_2P は，通信の立ち上がり時間がハイパーキューブの平均通信距離 $\log_2P$ に比例すると仮定して導入した．

$$\begin{aligned}&\gamma\frac{2N^3}{3PQ}+\beta_f\frac{N^2}{2P}+\beta_m\frac{3N^2}{2Q}\\&+a_f\log_2P\frac{N\left(1+NB\log_2P\right)}{NB}\\&+a_m\log_2P\frac{N\left(P-1+\log_2P\right)}{NB}\\&+b_f\frac{N\left(1+NB\log_2P\right)}{NB}+b_m\frac{N\left(P-1+\log_2P\right)}{NB}\end{aligned} \tag{4}$$

この式において N を固定し P と Q に対する基底関数が足りているかを検証する． N=40000 の場合を図 5 に示す．図は 1 ノード 24 スレッド並列，1 MPI プロセスとし，64 ノードのキューに，{P,Q}={1,64}, {2,32}, {4,16}, {8,8}, {16,4}, {32,2}, {64,1}と変化して観測データを測定した結果である．これらの図はこれらの基底関数が観測データを記述できることを示す．



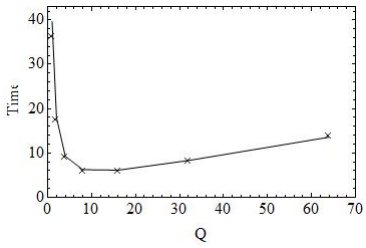


図 5 *N*=40000 の処理時間のモデル化　(図 3 ステップ 2)
（上：*P* を横軸にした場合，下：Q を横軸にした場合）

図 3 ステップ 3 により加えた基底関数は，ステップ 1 で得られた項を用いた作成した基底関数を 5 行目に加えた式(5)である．ここに $c_1 \leq 0$, $c_2 \leq 0$, $c_3 \leq 0$, $c_3 \leq 0$, c_0 は切片である．

ステップ 1 では-N の項を発見したが，それがどのようにプロセッサ数に依存しているかは，このように全ての場合を考慮して LP で変数選択することにより判明する．

$$
\begin{aligned}
T_{HPL}(P,Q,N) &= \gamma \frac{2N^3}{3PQ} + \beta_f \frac{N^2}{2P} + \beta_m \frac{3N^2}{2Q} \\
&+ a_f \log_2 P \frac{N\left(1+NB\log_2 P\right)}{NB} \\
&+ a_m \log_2 P \frac{N\left(P-1+\log_2 P\right)}{NB} \\
&+ b_f \frac{N\left(1+NB\log_2 P\right)}{NB} + b_m \frac{N\left(P-1+\log_2 P\right)}{NB} \\
&+ c_1 \frac{N}{PQ} + c_2 \frac{N}{P} + c_3 \frac{N}{Q} + c_4 N \\
&+ c_0
\end{aligned}
\tag{5}
$$

ステップ 4 では複数の *N* のデータで式(4)のモデルパラメータを LP により決定する．そこで N={40000, 50000, 60000} の観測データを加えて決定したモデルパラメータを表 3 に示す．この表は，モデルパラメータ γ, β_f, β_m, a_m, c_2, c_3, c_0 が選択されていることを示す．従ってモデル式は式(6)となる．

表 3　LP による変数選択とモデルパラメータ決定

γ	β_f	β_m	a_f	a_m	b_f	b_m
2.237 ・10^{-12}	1.706 ・10^{-8}	1.790 ・10^{-8}	0	3.757 ・10^{-4}	0	0

c_1	c_2	c_3	c_4	c_0
0	-9.081×10^{-5}	-1.741×10^{-3}	0	2.487

$$
\begin{aligned}
T_{HPL}(P,Q,N) &= \gamma \frac{2N^3}{3PQ} + \beta_f \frac{N^2}{2P} + \beta_m \frac{3N^2}{2Q} \\
&+ a_m \log_2 P \frac{N\left(P-1+\log_2 P\right)}{NB} \\
&+ c_2 \frac{N}{P} + c_3 \frac{N}{Q} + c_0
\end{aligned}
\tag{6}
$$

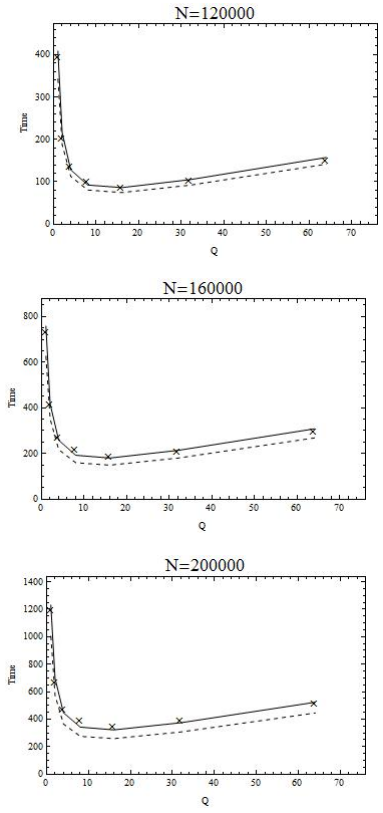


図 6　HPL の予測モデル(外挿)

図 6 の実線は N={40000, 50000, 60000}, {P, Q}={ {32,2},{16, 4},{8,8},{4, 16},{2, 32},{1, 64}}の 18 データを観測データとしてモデル化したときの N=120000, 160000, 200000 に対する予測である．実線は，式(6)のモデルによる予測で×の予測確認データと良く一致する．破線は式(6)から c_0, c_1, c_3 項を除いたモデルによる予測である．これらの図は，モデルパラメータ c_1, c_3 の基底関数及び切片 c_0 が予測に有効であることを示す．

4.2 EigenExa を京で実行した場合

EigenExa は実対称密行列の固有値計算をするプログラムで，次の 3 段階の計算ステップを持つ．

1) 行列の 5 重対角化
2) 5 重対角化行列の固有値・固有ベクトル計算
3) 固有ベクトルの逆変換

理化学研究所の京の 1 計算ノードは Fujitsu SPARC64™ Ⅷfx を搭載し、プロセッサ諸元は 8 コア/1 プロセッサ, 2GHz 動作である．主メモリは 16GB である．ノード間の通信は Tofu と呼ばれる 6 次元/メッシュトーラスで，論理帯域双方向 5GB/s, 隣接ノード間データ通信 20GB/s/ノードで実現している．計算ノードは 82,944 個あり，本報告では最大 8,192 のノードを使用した．EigenExa は Fujitsu c コンパイラでコンパイルし，Fujitsu SSL Ⅱ の BLAS を用いた．

本報告では EigenExa を京で実行し，ステップ 1 の行列の

5 重対角化の予測モデル構築を試みた結果を報告する．

図 7 は行列の 5 重対角化(PDD)の計算時間である．図は行列の大きさ N=30000 の同じ入力をストロングスケーリングで 6 回実行した結果である．黒の線が PDD 全体の計算時間．赤の線が MPI ライブラリの処理時間である．図は両曲線が滑らかでないことを示す．また p>1000 では実行毎に結果が異なることを示す．

図 8 は図 7 の黒の線から赤の線の時間の差，即ち PDD から MPI の処理時間を除いた時間である．6 つのデータは p>1000 でもよく一致しているので，図 7 に見られたこの領域の不連続性は MPI 由来であることがわかる．また図 7 と比べると p に対して滑らかなカーブとなる．そこで PDD の予測モデルの構築では，図 3 を適用して次の 3 つに分けて行うことにした．

ステップ 1）MPI の処理時間を除いた計算部をモデル化
ステップ 2）MPI 処理時間のモデル化
ステップ 3）ステップ 1 と 2 を用いた PDD のモデル化

ステップ 1）計算部のモデル化

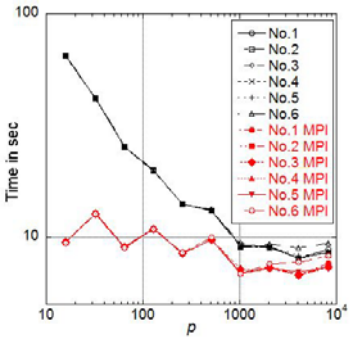


図 7　5 重対角化(PDD)の計算時間 (N=30000)

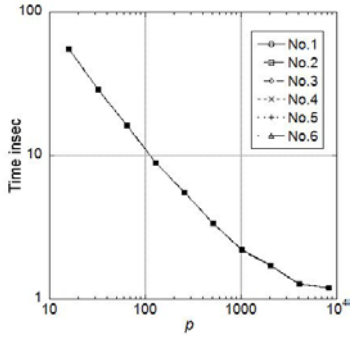


図 8　PDD の MPI を除いた計算時間 (N=30000)

図 9 は図 8 のように，PDD の MPI を除いた計算時間の N に対するスケーリングである．

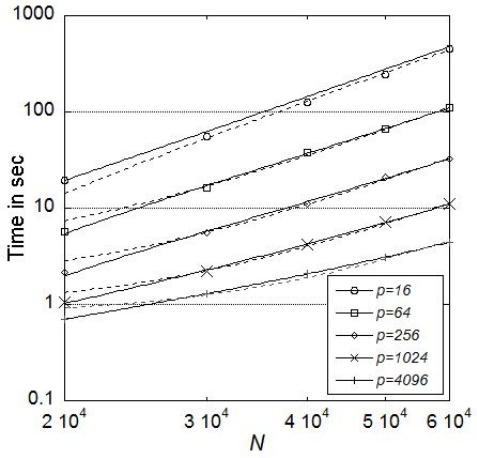


図 9　PDD 計算部の N に対する処理時間のモデル化

○，△，◇，×，＋は測定値である．破線は $c_1 \cdot N^3+c_0$ でモデル化したもので，測定値と微妙に一致しない．実線は $|c_1| \cdot N^3+ |c_2| \cdot N^2+ |c_3| \cdot N+c_0$ でモデル化したもので測定値とある程度一致する．従って図 3 ステップ 1 の結果として，この 3 つの基底関数を用いることにした．

図 10 は図 8 のように，PDD の MPI を除いた計算時間の p に対するスケーリングである．○，△，◇，×，＋は測定値である．破線は $c_1/p+c_0$ でモデル化したもので，p>1000 で測定値と一致しない．実線は $|c_1|/p+ |c_2|/\sqrt{p}+c_0$ でモデル化したもので測定値と一致する．従って図 3 ステップ 2 の結果としてこの 2 つの基底関数を用いることとした．

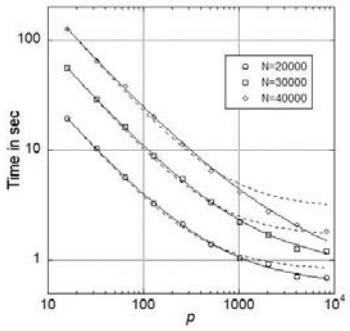


図 10　PDD 計算部の p に対する処理時間のモデル化

これらの結果から図 3 のステップ 3 として式(7)を用いることにした．

$$c_1\frac{N^3}{p}+c_2\frac{N^2}{p}+c_3\frac{N}{p}+c_4\frac{N^3}{\sqrt{p}}+c_5\frac{N^2}{\sqrt{p}}+c_6\frac{N}{\sqrt{p}} + c_7N^3+c_8N^2+c_9N+c_0 \quad (7)$$

この式と N={10000, 20000, 30000}, p={128, 256, 512, 1024, 2048, 4096, 8192}の観測データを用い LP により決定したモデルパラメータを表 4 に示す．この表により変数選択（モデルパラメータが 0 となった基底関数を除いた）モデル関数を式(8)に示す．

表 4　LP による変数選択とモデルパラメータ決定

c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_0
1.639 10^{-11}	3.093 10^{-7}	0	7.115 10^{-13}	1.357 10^{-8}	0	0	0	2.170 10^{-5}	0

$$c_1\frac{N^3}{p}+c_2\frac{N^2}{p}+c_4\frac{N^3}{\sqrt{p}}+c_5\frac{N^2}{\sqrt{p}}+c_9N \qquad (8)$$

式(8)を使った上記観測データのモデル化を図 11 に示す．実線が式(8)と表 4 を用いたモデル．破線が$\sqrt{p}$がない場合のモデル式によるモデルである．図はモデル式(8)が観測データを．より精密にモデル化したことを示す．

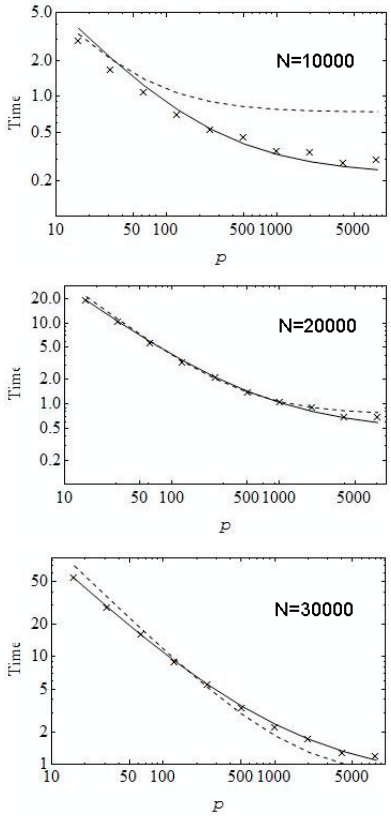
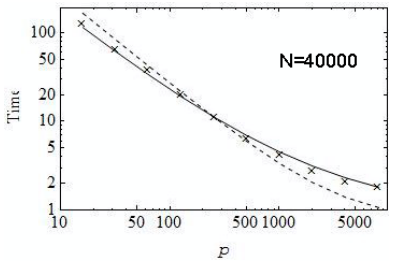


図 11　PDD 計算部の予測モデル(内挿)

図 12 は式(8)と表 4 を用い N={40000, 50000, 60000}を予測した結果である．図の実線はモデルが N に対して，p>1000 の領域を含めて，予測できることを示す．



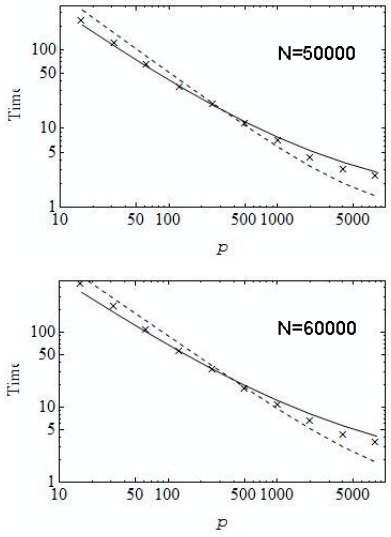


図 12　PDD 計算部の予測モデル(外挿)

ステップ 2）MPI 処理時間のモデル化

MPI 処理時間のモデル関数は式(9)のように仮定して LP により変数選択することにした．EgenExa は 2 次元のプロセスグリッドを採用しているので，通信は 1 次元の軸内に閉じている．従ってその通信量は，p の平方根に反比例し，また$\Sigma(N\text{-}k\text{+}1)$ (k=1, N-2)に比例する．これを式中 1 行目に示す．２行目では MPI_Bcast の通信量を考慮し，3 行目では MPI_Allreduce 等の MPI 中の演算を考慮した．

$$\begin{aligned}&c_{10}\frac{N^2}{\sqrt{p}}+c_{11}\frac{N}{\sqrt{p}}\\&+c_{12}N^2\log_2 p\\&+c_{13}N^3+c_{14}N^2+c_{15}N+c_{16}\end{aligned} \qquad (9)$$

モデル化のための観測データは N={10000, 20000, 30000}, p_{LB}={16, 64, 256, 1024, 4096}と p_{UP}={32, 128, 512, 2048, 8192}の２種類のデータを用いた．例えば N=30000 の p_{LB} は，図 7 の赤の鋸のような線の下側の包絡線のデータで，p_{UP} は上側の包絡線のデータである．このデータを用い LP で決定したモデルパラメータ表 5 に示す．

表 5　LP による変数選択とモデルパラメータ決定

	c_{10}	c_{11}	c_{12}	c_{13}	c_{14}	c_{15}	c_{16}
p_{LB}	1.295 $\cdot10^{-8}$	0	0	5.013 $\cdot10^{-14}$	0	1.932 $\cdot10^{-4}$	0
p_{UP}	3.847 $\cdot10^{-8}$	0	0	2.675 $\cdot10^{-14}$	0	2.235 $\cdot10^{-4}$	0

表 5 で零でない基底関数を選択し，MPI 処理部のモデル式として式(10)を得る．

$$c_{10}\frac{N^2}{\sqrt{p}}+c_{13}N^3+c_{15}N \qquad (10)$$

図 13 は観測データをモデル化した結果である．図中下の破線は LB，上は UB である．真ん中の実線は両モデルの

平均値である．図は *N*=20000 と 30000 がある程度平均値をモデル化していることを示す．

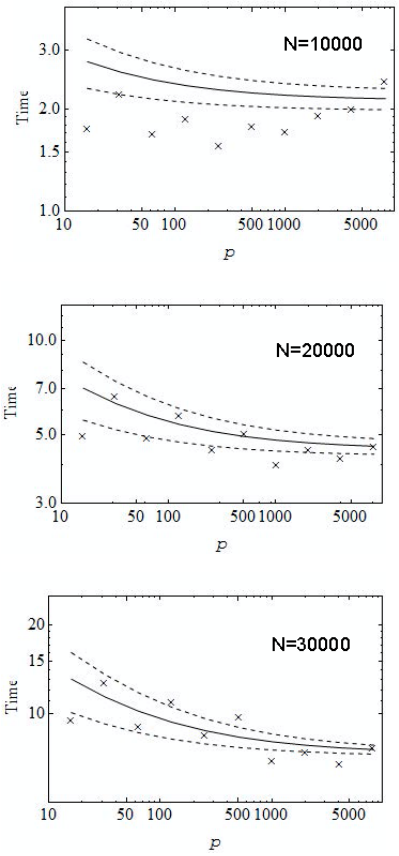
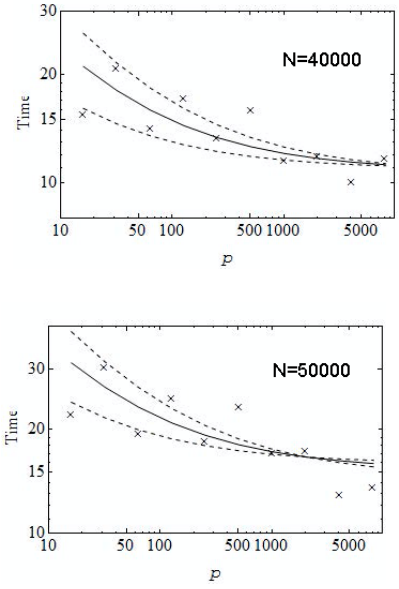


図 13　PDD の MPI 部のモデル化(内挿)

図 14 はモデルによる MPI 部の予測である．*N*=40000 は平均値をモデル化しているが，*N*=50000 と 60000 では p≧4096 で一致しなくなる．



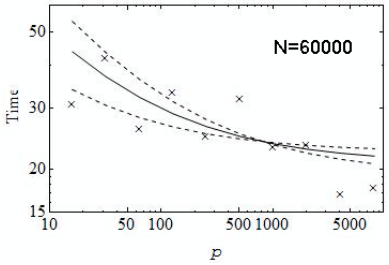


図 14　PDD の MPI 部のモデル化(予測)

ステップ 3）PDD のモデル化

これまでの PDD 計算部のモデル式(8), MPI 部のモデル式(10)より PDD 全体のモデル式(11)を得る．

$$c_1\frac{N^3}{p}+c_2\frac{N^2}{p}+c_4\frac{N^3}{\sqrt{p}}+(c_5+c_{10})\frac{N^2}{\sqrt{p}}+c_{13}N^3+(c_9+c_{15})N \qquad (11)$$

図 15 は PDD の観測データを LP でモデル化した結果で，もモデル関数は式(11)，モデルパラメータは，表 4 と 5 の非零値を除いたものと c_{10}, c_{13}, c_{15} は表 5 の p_{LB} と p_{UB} の平均値を採った，表 6 である．

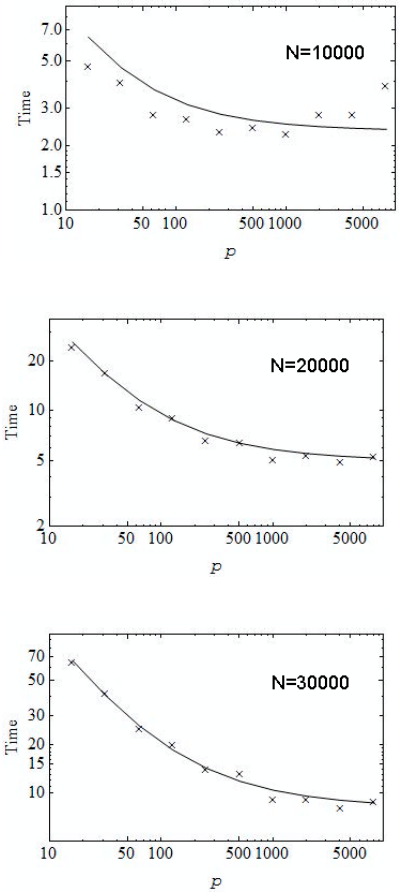


図 15 PDD のモデル化(内挿)

表 6　PDD のモデルパラメータ

c_1	c_2	c_4	c_5	c_9	c_{10}	c_{13}	c_{15}
1.639 $\cdot 10^{-11}$	3.093 $\cdot 10^{-7}$	7.115 $\cdot 10^{-13}$	1.357 $\cdot 10^{-8}$	2.170 $\cdot 10^{-5}$	2.571 $\cdot 10^{-8}$	3.844 $\cdot 10^{-14}$	2.084 $\cdot 10^{-4}$

図 16 はモデルによる予測である. ×印は確認データであり，モデルがこれらのデータを予測していることが確認できる.

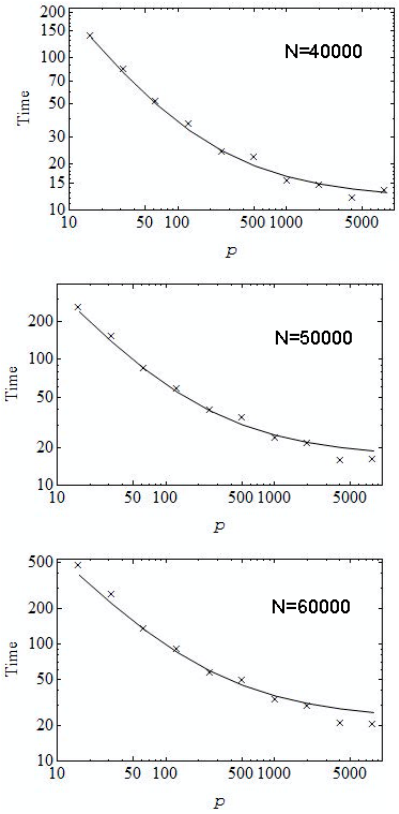


図 15 モデルによる PDD の予測

5. 議論とまとめ

HPL のモデル化において，負の値を持つ c_2, c_3 項はこれまでの経験からは予測できない基底関数であった. EigenExa の c_4 と c_5 の p の平方根に反比例する項もこれまでの経験からは予測できない基底関数であった．このような未知の現象に遭遇した場合，まず N に対するスケーリングモデル化，またストロングスケーリングに対するモデル化を行い, 基底関数の候補を見つけることが重要であった. また EigenExa の MPI 部のモデル化の場合のように，基底関数が予測できる場合は，N と p を変数とするモデル関数と観測データを用いて LP により変数選択を行うことにより，簡単にモデル関数を同定してモデル化を行えることを確認した.

今後は事例を増やし，実際の並列処理のモデル化に必要な方法を整理し，大規模並列計算の性能予測方法を確立していきたい.

本論文の結果（の一部）は，理化学研究所のスーパーコンピュータ「京」と，統計数理研究所のスーパーコンピュータ SGI ICE X を利用して得られたものです.

参考文献

[1] 折居茂夫，山本義郎：シンプレックス法を用いた非負のモデルパラメータを持つ並列処理時間モデルの予測力向上，情報処理学会論文誌，Vol. 56, No.6, pp. 1481-1495 (2015).
[2] Orii S. and Yamamoto Y., Improving Predictive Accuracy of Timing models for Parallel Processing by Using Non-negative Model Parameters, Proceedings of ISM HPCCON, October 9-10, 2015 in Tokyo.
[3] http://www.aics.riken.jp/labs/lpnctrt/EigenExa.html
[4] http://web.stanford.edu/~hastie/glmnet/glmnet_alpha.html
[5] http://www.netlib.org/benchmark/hpl/index.html