

スーパーコンピュータおよびPCクラスタにおける 自動チューニング機能付き並列固有値ソルバの性能評価

片桐 孝洋 (日本学術振興会特別研究員)

黒田 久泰 (東京大学情報基盤センター)

工藤 誠 (東京大学理学部情報科学科)

金田 康正 (東京大学情報基盤センター)

本稿では、スーパーコンピュータ および PC クラスタにおける自動チューニング機能付き並列固有値ソルバの性能を報告する。性能評価の結果、PC クラスタでのブロック化アルゴリズムの速度向上が自動チューニングにより 21 倍にも及ぶことが明らかになった。このことから PC クラスタ上で動作する並列固有値ソルバのチューニングでは、なんらかの自動化機能の付加が必須と考えられる。

Performance Evaluation of an Auto-tuned Parallel Eigensolver with a Super-computer and a PC-Cluster

TAKAHIRO KATAGIRI (JSPS Research Fellow) ,

HISAYASU KURODA (Information Technology Center, The University of Tokyo) ,

MAKOTO KUDOH (Department of Information Science, The University of Tokyo)

and YASUMASA KANADA (Information Technology Center, The University of Tokyo)

In this report, performance evaluation of a parallel eigensolver with an automatically tuning facility in a super-computer and a pc-cluster is reported. We found that the speed-up factor for a blocked algorithm on the pc-cluster was 21 times. The result indicates that the auto-tuning facility is needed to optimize blocked algorithms on pc-clusters.

1. はじめに

我々は現在、科学技術計算用ライブラリ群、特に並列数値計算において、以下に示す 4 つの特徴・機能を有する数値計算ライブラリの開発を行っている。(1) ユーザが指定するパラメタが少ないこと;(2) 演算カーネルに関する自動チューニング機能があること;(3) 並列計算機を用いる場合、通信処理に関する自動チューニング機能があること;(4) 利用するアルゴリズムに関する自動選択機能があること;今回我々はこの設計思想に基づく、固有値計算における直接法の自動チューニング機能を付加した並列数値計算プログラム集 ILIB (Intelligent LIBrary) の固有値ソルバに関する性能について報告する。

数値計算において自動チューニングを行うソフトウェアとして、PHiPAC¹⁾ や ATLAS²⁾ が有名であるが、これらのソフトウェアは逐次の BLAS を最適化するという局所最適化の方針をとる。したがって、(1) 本質的に並列処理を考慮していない;(2) 性能をさらに向上させるためには大域最適化が必須となる;それぞれの理由から最適化は不十分といえる。この一方で我々の開発した ILIB は、通信実装方式や通信粒度などの並列処理に必須な要因を最適化する。さらに複数の BLAS に影響を及ぼす要因も最適化をするという意味で、大域最適化を行うことが可能である。

2. 並列密対称標準固有値問題ソルバ

2.1 機能の説明

開発した並列固有値ライブラリは、式 (1) に示す標準固有値問題の固有分解を行うことができる。

$$AX = \Lambda X \quad (1)$$

ここで、係数行列 $A \in \mathbb{R}^{n \times n}$ は実数の対称密行列であり、固有値 $\lambda_i (i = 1, 2, \dots, n) \in \mathbb{R}$ から構成される行列を $\Lambda = \text{diag}(\lambda_i)$ 、固有ベクトル $x_i (i = 1, 2, \dots, n) \in \mathbb{R}^n$ から構成される行列を $X = (x_1, x_2, \dots, x_n)$ とする。式 (1) の固有分解を行う解法として Householder-二分-逆反復法を用いている。

今回自動チューニング機能を実装した部分は、(i) 対称実数密行列の相似変換における三重対角化ルーチン (ILIB_TrdRed); (ii) 修正 Gram-Schmidt 直交化ルーチン (ILIB_MGSA0); である。ILIB_TriRed ルーチンは、行列 A を三重対角行列 T に Householder 法を用いて相似変換をするルーチンである。一方、ILIB_MGSA0 ルーチンは、複数のベクトルを直交化させる (QR 分解する) ルーチンであり、固有ベクトルを求める際の逆反復法ルーチンで利用可能である。

2.2 自動チューニング機能

三重対角化ルーチン ILIB_TriRed における自動チューニングは、 k 反復時の演算処理の最適化に関して、(i) 行列-ベクトル積: $A^{(k)}u$; (ii) 行列更新:

$A^{(k+1)} = A^{(k)} - u(x^T - \mu u^T) - xu^T$ (ここで $u, x \in \mathbb{R}^n, \mu \in \mathbb{R}$); (iii) 通信処理に関する実装方式: (i) 行列-ベクトル積を実行する為に必要なベクトルのリダクション演算の実装方式; の最適化が行える.

一方, 直交化ルーチン ILIB_MGSAO は, ブロック幅 im でブロック化して実装されている. この場合のベクトルのデータ分散は列方向ブロック分割である. この処理における自動チューニングは, (i) 枢軸ブロック部の行列更新: $u_j = u_j - \mu_{ij}u_i$; (ii) それ以外の行列更新: $u_j = u_j - \mu_{ij}u_i$; (iii) 通信粒度に関連するブロック幅 im_i において最適化がなされる.

3. 性能評価

この章では, スーパーコンピュータ HITACHI SR2201 および PC クラスタを用いて評価した結果を記す. 本評価で使用した SR2201 の各ノードの理論ピーク性能は 300MFLOPS, ノード間は三次元ハイパクロスバ網で結合されており, その最大転送性能は 300Mbyte/秒である. またコンパイラとして, 日立の最適化 FORTRAN90 V02-06-/D, オプションとしては `-rdma -W0, 'OPT(O(SS))'` を指定した. 一方 PC クラスタのノードとして, Intel PentiumIII 800MHz Dual を 4 ノード用いた. ノード当たりの搭載メモリは 512MB, ネットワークカードは PCI 10/100Mbps Intel EtherExpressPro100+ である. 使用 OS は Linux 2.2.13-33smp である. またコンパイラとして, PGI 社の Fortran90 コンパイラ 3.2-3, オプションとしては `-fast` を指定した. 通信ライブラリとしては, 双方とも MPI (Message Passing Interface) を利用した.

3.1 自動チューニングの結果

ILIB_TriRed での自動チューニングのパラメタは, (I) 行列-ベクトル積のアンローリング (TMatVec) = { なし, 2 段, 3 段, ..., 8 段, 16 段 }; (II) 行列更新のアンローリング (TUpdate) = { なし, 2 段, 3 段, ..., 8 段, 16 段 }; (III) 通信方式 (Tcomm) = { Tree, MPI }; の合計 3 種類である. (I), (II) の処理は BLAS2 演算である. ここで (III) の "Tree" とは, 1 対 1 通信を用いた二分木通信形態でベクトルリダクションを実現する方式であり, "MPI" とは MPI の関数 `MPI_ALLREDUCE` を利用する方式である.

一方, ILIB_MGSAO での自動チューニングのパラメタは, (I-1) 枢軸ブロック更新処理の最外ループのアンローリング (MBOuter) = { なし, 2 段, 3 段, 4 段 }; (I-2) 枢軸ブロック更新処理の第 2 ループのアンローリング (MBSecond) = { なし, 2 段, 3 段, ..., 8 段, 16 段 }; (II-1) それ以外の更新処理の最外ループのアンローリング (MOOuter) = { なし, 2 段, 3 段, 4 段 }; (II-2) それ以外の更新処理の第 2 ループのアンローリング (MOSecond) = { なし, 2 段, 3 段, ..., 8 段, 16 段 }; (III) ブロック幅 (MBlklen) = { 1, 2, ..., 6, 8, 16 }; の合計 5 種類である. (I-1)-(II-2) の処理は

BLAS3 演算である.

3.2 実験結果

表 1 に 4 ノードでの自動チューニング結果を示す. なお表中の括弧のパラメタは, ILIB_TriRed においては (TMatVec, TUpdate, Tcomm) であり, ILIB_MGSAO においては (MBOuter, MBSecond, MOOuter, MOSecond, MBlklen) である.

表 1 スーパーコンピュータおよび PC クラスタでの実験結果
(a) HITACHI SR2201 での自動チューニング結果

処理	パラメタ固定	最適化	速度向上
ILIB_TriRed (8000 次元)	3589 [sec] (8,6,Tree)	1962 [sec] (3,2,MPI)	1.8
ILIB_MGSAO (4000 次元)	892.7 [sec] (4,8,4,8,4)	420.2 [sec] (1,1,1,3,6)	2.1

(b) PC クラスタでの自動チューニング結果

処理	パラメタ固定	最適化	速度向上
ILIB_TriRed (4000 次元)	308.6 [sec] (8,6,Tree)	301.6 [sec] (6,16,Tree)	1.02
ILIB_MGSAO (2000 次元)	591.2 [sec] (4,8,4,8,4)	28.06 [sec] (4,5,2,4,8)	21.0

4. 考察とまとめ

表 1 から BLAS2 演算の自動チューニングに関して, SR2201 の方が PC クラスタよりも効果があることが分かる. この理由は, PC クラスタでは演算時間に対しメモリアクセス時間が大きいことによるものと考えられる. すなわち BLAS2 演算では BLAS3 演算に比べ, メモリアクセス量/演算量の比が大きいので, このような計算機では性能向上が見込めない.

この一方で PC クラスタでは, ブロック化アルゴリズムの主演算 (BLAS3) に対して 21 倍も高速化される. この結果も同様に PC クラスタでは, メモリアクセス時間の隠蔽が高速化においていかに重要かを示している. またこの結果は, パラメタ固定の実装方式は性能の観点から回避すべきであることも示唆している. 以上の理由から, 本稿で示したような自動チューニング機能が PC クラスタ上でのブロック化アルゴリズムにおいて, 特に効果的であることが期待される.

なお ILIB に関する情報は, <http://www.hints.org/> から入手可能である.

参考文献

- 1) Bilmes, J., Asanović, K., Chin, C.-W. and Demmel, J.: Optimizing Matrix Multiply Using PHiPAC: a Portable, High-Performance, ANSI C Coding Methodology, *Proc. International Conference on Supercomputing 97*, pp. 340-347 (1997).
- 2) Whaley, R. C. and Dongarra, J. J.: Automatically Tuned Linear Algebra Software, ATLAS project, <http://www.netlib.org/atlas/index.html>.