ナノ材料第一原理分子動力学プログラム PHASE の京速コンピュータ「京」上の計算性能最適化

黒田明義† 長谷川幸弘† 寺井優晃† 井上俊介† 市川真一‡ 小松秀実‡

大井憲行§ 安藤琢也§ 山崎隆浩** 大野隆央†† 南一生†

ナノ材料第一原理分子動力学プログラム「PHASE」を対象に京速コンピュータ「京」向けの計算性能最適化を実施した．本アプリケーションで用いられる密度汎関数法における収束計算には最急降下法や共役勾配法などが使われ，直交化や対角化の処理を含む．また波動関数をフーリエ級数展開するため FFT 計算が用いられている．これらは系の大きさに対して最大 3 乗の演算量を持つため，計算方法の工夫により大幅な性能向上が見込まれるが，計算サイズの拡大は困難であり超並列化が難しいとされている．本論文では，大規模計算にて画期的な研究成果をあげることを目的に行った計算性能最適化について，その手法を紹介する．また実測評価にて並列性能が向上し，20%程度の実行効率達成の見込みを得たので報告する．

Performance Optimization of First-Principles Molecular Dynamics Simulator for Nanomaterials "PHASE" for the K computer

AKIYOSHI KURODA†, YUKIHIRO HASEGAWA†, MASAAKI TERAI†,
SHUNSUKE INOUE†, SHIN-ICHI ICHIKAWA‡, HIDEMI KOMATSU‡,
NORIYUKI OHI§, TAKUYA ANDO§, TAKAHIRO YAMASAKI **, TAKAHISA OHNO††
and KAZUO MINAMI†

We optimize the first-principles molecular dynamics simulator for nanomaterials "PHASE" targeting the K computer. PHASE performs convergence calculations using the steepest descent and the conjugate gradient methods, which are based on orthogonalization and diagonalization. Also, FFT is used in order to expand wave functions with the Fourier series. These calculations are $O(N^3)$ for a given system size N. For this reason, we expect to increase sustained performance largely by improving the calculation methods. However, it has been said that it is difficult to achieve massive parallelism because calculation time diverges in large systems. In this paper, we introduce optimization methods developed to achieve outstanding research results with large scale computations. By doing actual performance measurements with the K computer, we confirm that the methods increase parallel performance and achieve sustained performance of 20 % or higher.

† 理化学研究所 次世代スーパーコンピュータ開発実施本部 RIKEN, Next-Generation Supercomputer R&D Center
‡ 富士通株式会社 FUJITSU, LTD.
§ 株式会社富士通長野システムエンジニアリング FUJITSU NAGANO SYSTEMS ENGINEERING LTD.
** 株式会社富士通研究所 FUJITSU LABORATORIES LTD.
†† 物質・材料研究機構 理論計算科学ユニット NIMS, Computational Materials Science Unit (CMSU)

1. 序章

理化学研究所では，10PFLOPS 級のスーパーコンピュータとなる京速コンピュータ「京」(以下「京」，と呼ぶ)を平成 24 年度秋の共用開始を目指し開発中である．この「京」の性能を検証し，これまでの計算機では不可能であったような大規模計算を行うことで画期的な研究成果をあげるべく，いくつかの実アプリケーションを選定し，計算性能最適化を施す作業を行っている．

本論文では，こうして選定されたアプリケーションの 1 つである擬ポテンシャルと密度汎関数法によるナノ材料第一原理分子動力学プログラム(以下，PHASE と呼ぶ)[1]に対し，コードを解析し，「京」向けに行った計算性能最適化並びにその成果について報告する．1節では，計算性能最適化の概要を説明し，背景となる技術について議論する．2節では，実際に PHASE で行われた計算性能最適化手法について説明する．3節ではこの最適化に対する実測による評価結果を説明し，4節では性能評価結果について考察する．

1.1 PHASE とは

PHASE は，擬ポテンシャルと密度汎関数法によるナノ材料第一原理分子動力学プログラムであり，局在基底ではなく平面波基底を用いることにより，分子から固体まで多くの物質に対して高精度な電子状態計算が可能である．

本アプリケーションは特に結晶・アモルファス状態などの電子状態計算を得意とし，現象をミクロなスケールから量子力学的に厳密に解明し，新規材料開発につながる計算を目指す(図 1)．「京」でターゲットとする計算対象としては半導体素子のデバイス特性解析や燃料電池の伝導度計算などによるエネルギー問題があげられる．

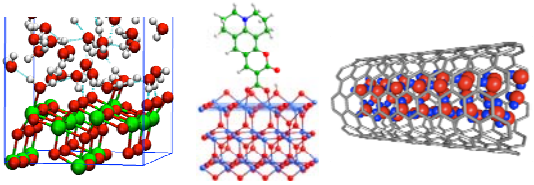

図 1 PHASE を用いた計算例．左図は酸化物・水界面の分子動力学解析．中図は色素分子表面吸着系の光吸収スペクトル解析．右図はカーボンナノチューブ中の水分子の構造解析．

本手法は，物質の電子状態について量子力学を用いて厳密に解析するため，Schrödinger 方程式を扱うことになる．

$$\left\{-\frac{\hbar^2}{2m}\nabla^2+V(\mathbf{r})\right\}\Psi_i(\mathbf{r})=E_i\Psi_i(\mathbf{r}) \qquad (1)$$

ここで，多粒子の波動関数 Ψ_i を，粒子によらない 1 粒子密度関数で記述することで，多体粒子問題を平均場中の 1 電子問題に置き換えることが可能である．この手法を密度汎関数法と呼び，Kohn-Sham 方程式に帰着される[2-3]．

$$\left\{-\frac{\hbar^2}{2m}\nabla^2+V(\mathbf{r})+V_{XC}[\rho(\mathbf{r})]\right\}\varphi_i(\mathbf{r})=\varepsilon_i\varphi_i(\mathbf{r}) \qquad (2\text{-}1)$$

$$\rho(\mathbf{r})=\sum_i|\varphi_i(\mathbf{r})|^2 \qquad (2\text{-}2)$$

ここで V_{XC} は交換エネルギーで，$\rho(\mathbf{r})$ は電子電荷密度，φ_i は波動関数である．この解を求めるためには，離散化が必要となり，本手法では波動関数をフーリエ級数展開し，その展開係数を求める方法を用いる[4-9]．

$$V=V_H+V_{PP-Local}+V_{NL} \qquad (3\text{-}1)$$

$$V_H(\mathbf{G})=4\pi\frac{\rho(\mathbf{G})}{\mathbf{G}^2} \qquad (3\text{-}2)$$

$$V_{PP-Local}(\mathbf{G})=\sum_I V^{s(I)}_{PP-Local}(\mathbf{G})\exp(-i\mathbf{G}\cdot\mathbf{R}_I) \qquad (3\text{-}3)$$

$$V_{\mathrm{NL}}=\sum_I\sum_{s,t}D^I_{st}\left|\beta^I_s\right\rangle\left\langle\beta^I_t\right| \qquad (3\text{-}4)$$

$\boldsymbol{G}$ は波数，β^I_s は原子 I と(l, m, τ)の組み合せ s についての擬ポテンシャル成分である．(l は方位量子数，m は磁気量子数，τ は参照エネルギーを表す．) 波数展開係数を求めるには，離散化されたハミルトニアン行列から固有状態ベクトル束 Ψ_i を計算したのち，密度場 $\rho(\mathbf{r})$ を経由して，交換エネルギー V_{XC} を構築するという一連の作業を，自己無撞着な密度場に収束するまで繰り返し計算を行う．この収束計算を Self Consistent Field 法(SCF)と呼ぶ．収束方法には，共役勾配法(CG)や修正最急降下法(MSD)，最少残差法(RMM)，安定性にすぐれた Davidson 法などを収束条件によって選択可能で，波動関数の直交条件を保持するために Gram-Schmidt 直交化や対角化などの手続きが反復計算の中に含まれる．

平面波基底を用いた密度汎関数法の並列化については 20 年程前から多くの研究があり[10-13]，フーリエ変換の最適化や BLAS Level3 の適用について議論されてきた．BlueGene/P 上では，65,536 並列まで高性能を発揮できることが実証されている[13]．また PHASE についての，同システムでの最適化事例については，HPCS 2009 にて紹介されている[14]．

1.2 評価システム

性能最適化の評価には現在開発中の「京」並びに同等のアーキテクチャのコンピュータを用いて行った．「京」とは，2012 年秋に理化学研究所計算科学研究機構にて共用開始予定のスーパーコンピュータシステムである．国家基幹技術として，10PFLOPS 級のスーパーコンピュータを開発することで，スーパーコンピューティング技術を維持発展させるとともに、わが国の共用施設として幅広い計算科学分野の利用に供する世界最高のコンピューティング基盤を整備することを目的としている．

計算に用いるノードは，1CPU, 16GB のメモリ，ノード間のデータ転送を行うインターコネクト用 LSI（ICC: Inter-Connect Controller）で構成されている．CPU は「京」向けに新規に開発された富士通社製の SPARC64™ VIIIfx[15-19]であり，1 CPU 上に 8 コアの演算器を持つ．ピーク性能は 128 GFLOPS で，メモリバンド幅は 64GB/s (0.5B/F) である．浮動小数点レジスタは 256 本に拡張され，コンパイラによる命令スケジューリングが容易となっている．また SIMD 命令の導入による，ベクトル計算，マスク演算が可能となり，コンパイラにより後者はプログラム中の分岐での命令スケジューリングの向上につながる．更にセクタキャッシュ機能が導入され，一部再利用性のあるデータを留め置くことが可能となり，キャッシュからデータが溢れることで性能低下するアプリケーションでの効果を期待できる．スレッド並列についてはコア間の並列処理のための同期をとるためのハードウェアバリア機構を備えることで，ノード内のスレッド並列効率が飛躍的に向上している．

計算に用いるノードは，Tofu インターコネクトと呼ばれる 6 次元メッシュ/トーラスネットワークで結合されている[20-21]．Tofu インターコネクトは，3 次元メッシュ/トーラスネットワークと基本単位と呼ばれる 2x3x2 のメッシュ/トーラスネットワークを組み合わせて構成されている．各ノードからは 10 本のリンクが出ており，6 本が 3 次元メッシュ/トーラスに，残りの 4 本は基本単位の内部結合に用いられている．各リンクのバンド幅は 5GB/s(双方向)で，システム全体のバイセクションバンド幅は 30TB/s である．Tofu インターコネクトでは，このリンクの冗長性を活用してジョブ単位で 3 次元トーラスネットワークを構成することが可能で，更に故障ノードがあっても 3 次元トーラスを維持することができ，利便性だけでなく信頼性も高いネットワークとなっている．

システムは，1 枚のシステムボードに 4 個のノードが搭載され，1 台の筐体には 24 枚のシステムボードが搭載されている．「京」ではこの筐体が 864 台設置される．ピーク性能 10.62PFLOPS，全メモリ量 1.26PB である．

OS は Linux をベースとしており，ファイルシステムはグローバルファイルシステムからローカルファイルシステムへステージングする運用となる．プログラミング言語は Fortran，C，C++が用意され，スレッド並列には自動並列や OpenMP が利用可能である．プロセス並列には MPI-2.1 仕様に準拠するメッセージ通信ライブラリが用意されてい

表 1「京」を用いた評価システム

ハードウェア
SPARC64™ VIIIfx, 2GHz, 8 core/CPU, 1CPU/node 浮動小数点演算性能：128GFLOPS/node 浮動小数点レジスタ：256 本/core キャッシュ：L1 - 32KB/core, L2 - 6MB/CPU メモリ：16GB/CPU, 0.5B/F ネットワーク：3D torus, 5GB/s 6 方向×双方向
ソフトウェア
・Linux ・「京」向け言語開発環境(Fortran，MPI, SSL2(BLAS, LAPACK，ScaLAPACK)を含む) ・FFTW

表 2 富士通 FX1 を用いた評価システム

ハードウェア
SPARC64 ™ VII, 2.5GHz 4core/CPU, 1CPU/node 浮動小数点演算性能：40GFLOPS/node 浮動小数点レジスタ：32 本/core キャッシュ ：L1$ - 64KB/core，L2 - 6MB/CPU メモリ ：32GB/CPU, 1B/F ネットワーク：Fat-tree, 2GB/s 双方向
ソフトウェア
・Open Solaris ・富士通 Parallelnavi（Fortran，MPI，SSL2（BLAS, LAPACK, ScaLAPACK）を含む) ・FFTW

る．「京」では並列手法として，ノード内はスレッド並列，ノード間はプロセス並列というハイブリッド並列を推奨しているが，スレッド並列は現在組み込み中のため，本評価では 1 ノードあたり 1 コアのみにプロセスを割り当てての測定結果を使用した．科学技術ライブラリは，BLAS, LAPACK, FFTW 並びに富士通数値演算ライブラリ SSLII が利用可能である．評価環境は**表 1**の通りである．

同等なアーキテクチャの評価環境としては，富士通社製の FX1 を，「京」が使える以前の先行評価機として利用した．比較のため FX1 を用いた評価条件を**表 2**にまとめる．

1.3 性能最適化の手順

「京」上で実施したアプリケーションの計算性能最適化は，並列計算による性能向上と CPU 単体内での性能向上の両面から行うとし，以下の手順で進めている．

(1) 並列特性分析
(2) カーネル抽出と評価
(3) 性能最適化方法の選択
(4) 性能最適化の試作
(5) 実機における性能評価

まずオリジナルアプリケーションに対して，アルゴリズム毎に処理ブロック(計算/通信)に分割し，ブロックごとに並列性能を実測する．但し，ブロックとは，サブルーチンよりも細粒度(ループレベル)を考える．複数のサブルーチンに同じような処理(ループ構造)が含まれていれば，同種のブロックとして扱う．測定結果から，大規模計算時に高コストとなる箇所を見積もり，カーネルとして抽出する．抽出後はカーネルごとに，並列性能や単体性能の阻害要因を見積もり，対応策を検討する．選択された性能最適化方法について試作を行い，実機を用いてその並列性能，単体性能の観点から効果を評価した上で，最終的にアプリケーションにフィードバックする．

1.4 抽出されたカーネル

本アプリケーションは多様な材料の電子状態計算を行うため，汎用性を重んじ，様々な計算手法が選択可能である．このため性能最適化の対象カーネルも多岐にわたる．1.3節で述べたカーネルとして抽出したものはSCFループ中の以下11区間である．

- 区間 1: V_{local}の逆 FFT
- 区間 2: $V_{nonlocal}$を波動関数 ϕ_i と βの内積 f に作用
- 区間 3: V_{local}を波動関数 ϕ_iに作用，波動関数の修正値 $H\phi_i$を計算
- 区間 4: $f_{ijt}=\beta \cdot \phi$の計算
- 区間 5: Gram-Schmidt の直交化
- 区間 6: 固有値計算，波動関数 ϕ_i と f_iのバンド方向並べ替え
- 区間 7: 電荷密度計算
- 区間 8: V_{local}の逆 FFT
- 区間 9: 行列対角化計算，波動関数 ϕ_iの修正
- 区間 10: $f_{ijt}=\beta \cdot \phi$の計算
- 区間 11: 電荷密度，ポテンシャル，全エネルギー計算

以上のカーネルを分析すると，主に行列-行列積の形に書き換え可能な区間，FFT を含む区間，対角化を含む区間の 3 種類に分類が可能である(**表 3**)．単体性能の面から性能阻害要因を見積もると，行列-ベクトル積で記述されているため，性能がメモリバンド幅律速のものが挙げられ，並列性能の面からは，並列粒度が細かくなることによる性能低下，FFT の並列性能，対角化の並列性能の 3 点が考えられる．以上の 4 つの観点に着目して，それぞれ性能最適化手法を検討し，試作，評価を実施した．

表 3 PHASE の 11 区間のカーネルの分類

種類	区間番号
行列-行列積に書き換え可能	2,4,5,8,9,10
FFT を含む	1,3,6,7,8,11
対角化	9

2. 計算性能最適化手法

1.2節で述べた「京」の仕様の中で，PHASE の性能に影響すると考えられるものは，メモリバンド幅が 0.5B/F あること，並びに，並列数が 8 万以上と超並列であることである．本節ではこれらの仕様を踏まえて1.4節で見積られた 4 つの性能阻害要因について用いた性能最適化手法について説明する．

2.1 BLAS Level3 適用

一般に，密度汎関数法は，ハミルトニアン行列に対する固有値問題であるため，系の規模 N に比べて演算量が大きく $O(N^3)$である．これは，計算に必

要なデータ量に比べて演算量が大きいことを意味し，計算を工夫することにより，メモリ帯域を圧迫しない実装が可能である．単体性能向上策として，演算密度が高いカーネルの演算効率を上げる工夫として，BLAS Level3 の適用について説明する．実際には，

$$C_{ij} = \sum_k A_{ik} B_{kj} \tag{4}$$

という行列-行列積の形で記述された箇所が対象となる．これらのカーネルでは，行列 C_{ij} の計算に A_{ik}, B_{kj} の要素数 $2N^2$ 個のデータを用いて，$2N^3$ 回の演算を行う．1 回の参照で多数演算を行うため，メモリへ何度もデータを読み込む必要がなく，演算密度が飛躍的に向上する可能性がある．

カーネル区間 5 の Gram-Schmidt 直交化では，直交ベクトル v_j を得るために，オリジナルベクトル(x_j)，直交ベクトル束($v_1 \sim v_{j-1}$)を用いて，

$$v_j = \frac{x_j - \sum_{i=1}^{j-1}(x_j \cdot v_i)v_i}{\left\| x_j - \sum_{i=1}^{j-1}(x_j \cdot v_i)v_i \right\|} \tag{5}$$

という計算を行う．このため新しい直交ベクトルを得るために，それまでに生成した全直交ベクトル束が必要となり，依存関係のため直交ベクトル束を一度に計算することは出来ない．一般的には直交ベクトルを生成するために，行列-ベクトル積の形で逐次に計算されることが多いが，ある程度の世代($i=1 \sim m$)まで直交ベクトルが生成できていれば，

$$v_j = \frac{x_j - \sum_{i=1}^{m}(x_j \cdot v_i)v_i - \sum_{i=m+1}^{j-1}(x_j \cdot v_i)v_i}{\left\| x_j - \sum_{i=1}^{m}(x_j \cdot v_i)v_i - \sum_{i=m+1}^{j-1}(x_j \cdot v_i)v_i \right\|} \tag{6}$$

と，依存関係のある部分とない部分を分離することが可能である．分子第 3 項は，そのまま依存性として残存するが，分子第 2 項は依存関係が消えるため，ベクトルを m 行束ねる(ブロック化)ことで，行列-行列の形で書き下すことが可能である[22].

また非局在ポテンシャル V_{NL} の計算を例にとると，式(3-4)から，

$$V_{\mathrm{NL}} = \sum_I \sum_{s,t} D_{st}^I \left|\beta_s^I\right\rangle\left\langle\beta_t^I\right| \tag{7}$$

となっている．これらの計算は縮約の形式で記述されるため，計算順序の変更などで行列-行列積の形で置き換え可能である．

2.2 2 軸並列化

密度汎関数法では，演算量が系の規模 N に比べて $O(N^3)$であり，SCF による反復計算も加わるため，演算量が非常に多い手法である．このため超並列システムでは，メモリ量や演算時間の制限から，Nの大きな計算が困難となる．N を大きく出来ないと，並列粒度が細かくなるなどの問題が新たに生じ，極端なケースでは N が並列数を下回り，それ以上の分割が出来ない事態も生じ得る．しかし，コーディングの観点から見ると，プログラム中のループ構造は N のループだけではなく，他の変数のループを含む多重ループ構造となっている．これは N 以外のループでの並列化も可能であることを意味し，これを多軸並列化と呼ぶ．

PHASE での並列方法はカーネルごとに異なるが，多くはエネルギーバンドのループを用いて並列化(1 軸並列)されている．しかし，各エネルギーバンドは，波動関数のフーリエ級数展開により生成した波数方向の自由度を持ち，その方向に並列軸を拡大(2 軸並列)することで，並列効率の向上が見込まれる(図 2).

カーネル区間ごとにバンド並列と波数並列が混在していると，並列軸の入れ替えに伴いデータの持ち替えが必要となり，全体通信が発生する．しかし 2 軸並列化により，これらの軸の入れ替えに伴う全体通信を局所化することが可能となり，通信の削減効果も見込まれる．

カーネル区間 5 の Gram-Schmidt 直交化については，精度は高いが，依存関係がより大きい，修正 Gram-Schmidt 法に対して，2 軸並列化を行った．本区間は，その他の区間と異なり，並列数の制約から波数方向に並列化を行っていたが，超並列時には，式(5)からバンド方向に大量のデータ縮約通信が多数回発生する．2 軸並列化を行うことで，バンド方向へ並列軸の拡大により，この縮約通信の並列効果が期待できる．しかし，2.1節で述べた BLAS Level3 適用に伴う，バンド方向のブロック化と干渉し，ブロックサイズを大きくするのが難しくなることによる性能への影響や，依存関係のある箇所により引

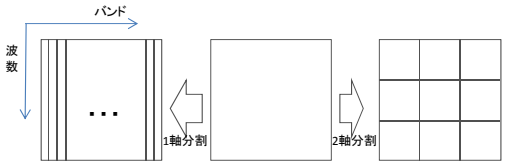

図 2 一軸分割と 2 軸分割，波数方向に分割軸を増やすことで，バンド方向の粒度が大きくなる．

き起こされるロードインバランスなどが問題となる．本手法では，計算順序を工夫することで，バンド方向の分割による性能への影響を波数方向の分割でカバーし，高い並列性能を実現している[23]．

2.3 FFT の高速化

本アプリケーションはフーリエ級数を基底関数として用いているためフーリエ変換を使用する．フーリエ変換は高速フーリエ変換(FFT)を採用することで，超並列時の実行時間は行列-行列積の箇所に比べて負荷割合は増大しにくい．しかし並列 FFT の転置計算の際に全体通信が発生し，これが並列数と共に増大する傾向にある．ここで，2.2節で説明した 2 軸並列を考慮すると，並列 FFT は全ノード間全体通信の必要がなく，ある並列軸方向の通信を行うことで十分であることが分かる．これは通信相手が減るだけでなく，通信マッピングの最適化により経路競合も防げるため，1 軸並列を用いた並列手法に比べて大幅に通信時間を削減が可能となる(図3)．「京」全体で FFT を行うと，システム全体で80,000 程度のノード間の全体通信が発生するが，2 軸並列により，300 程度のノード間の全体通信に縮小が可能となる意義は大きい．またこれらの FFT はバッファリングによりある程度まとめて処理することも可能であり，メモリ使用量に余裕がある場合，通信回数を削減することも可能である．

2.4 対角化計算

密度汎関数法の中心となる計算は，波動関数の収束計算であるが，精度を保ちつつ，収束を高速化するために，直交化や対角化が使用され，対角化には Householder 法など精度を要求するアルゴリズムが用いられている．直交化は BLAS level3 適用で高速化できたが，これらの対角化手法は Householder 変換として行列ベクトル積の部分が残存し，加えて並列効率が低いという問題が残る．

一般に，対角化の対象となる行列の規模は，原子

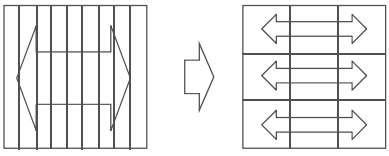

図 3 2 軸並列化により，通信をグルーピングすることで通信相手が減り，通信経路の輻輳も削減できる．

表 4 Gram-Schmidt 直交化(区間 5)の BLAS Level3 適用事例. 網掛けは BLAS Level3 を適用可能箇所で，太字が適用済みの箇所である．

サブルーチン	時間 [sec]	比率 [%]	演算効率 [%]
区間4	512.7	100.0	28.23
m_ES_F_transpose_r	105.3	20.5	0
m_ES_W_transpose_r	15.7	3.1	0
WSW_t	15.2	3.0	0.036
normalize_bp_and_psi_t	0.9	0.2	3.25
W1SW2_t_r	49.2	9.6	5.46
modify_bp_and_psi_t_r	50.8	9.9	4.45
W1SW2_t_r_block	**162.0**	**31.6**	**41.89**
modify_bp_and_psi_t_r_block	**96.2**	**18.8**	**74.65**
m_ES_W_transpose_back_r	14.1	2.8	0
m_ES_F_transpose_back_r	1.3	0.3	0

数の数倍程度と比較的小規模なものである．アプリケーションを実行する際に用いられる実問題では，全空間対角化を必要とはしておらず，部分空間での対角化を行っているため，小規模環境での計算では非並列の LAPACK を用いてきた．しかし「京」上での計算では問題規模の増大を想定し，ScaLAPACK を導入して，対角化箇所の並列化を行った．しかし ScaLAPACK は，並列性能に限界があることが知られており，より並列性能の良い対角化手法が必要とされる．「京」では，高並列対角化ライブラリの開発を共同で進めており[24-25]，現在そのライブラリの導入も試みている．

3. 評価結果

本節は2節で説明した性能最適化手法を適用し，実測による評価結果を説明する．

3.1 BLAS Level3 の適用の評価結果

カーネル区間 5 の Gram-Schmidt 直交化における BLAS Level3 適用結果について報告する．表 4 は FX1 16 並列で測定した $HfSiO_2$ 768 原子アモルファス系の SCF6 回分の実行結果である. 網掛けの箇所は BLAS Level3 適用可能箇所である．網掛けの箇所のうち太字の 2 つのサブルーチンはブロック化済みの箇所で，演算効率 40～70%となっている．それに対して，細字の 2 サブルーチンは 4%～5%の演算効率である．Gram-Schmidt 直交化は，もともと依存関係のある計算のため一部消去できないオリジナルプログラムの依存関係が行列-ベクトル積の形で残っており，細字の 2 サブルーチンがこれに相当する. その箇所の効率が 4%～5%程度で

あるということは，ブロック化する前の効率も同等の 4%～5%とみなせるので，BLAS Level3 適用箇所は 8 倍～15 倍程度性能が向上しているのが分かる．区間 5 全体で見ても効率が 4～5%から 28%へと 5 倍以上性能が向上したことになる．

BLAS level3 の性能は，行列のブロックサイズが大きいほど効率が良くなる．しかしこのサイズを大きくすると，依存関係のある箇所の割合も大きくなり，演算性能の低下の原因にもなる．さらにこのサイズが大きいと，並列性能の確保も困難になり，これらの効果を考慮して，行列のブロックサイズを決める必要がある．

3.2 2 軸並列化の評価結果

図 4は3.1節で用いた系 384 原子を用いて，カーネル区間 2 の非局所ポテンシャル計算カーネルの 2 軸並列化の効果について，FX1 上での並列特性を比較したグラフである．左のグラフは 1 軸並列時の実行時間で右のグラフは 2 軸並列化後のグラフである．2 軸並列の方はバンド方向の分割数を 2 に固定して測定を行った．1 軸並列時は，128 並列で既に並列効率が悪化しており，経過時間の下げ止まりが見られるが，2 軸並列版では，分割軸が増えたことに伴う新規通信が多少発生しているが，128 並列でも良好な並列効率を保っている．

カーネル区間 5 の Gram-Schmidt 直交化においても，同様の並列効果が得られている．3.1節で述べた行列のブロックサイズにより，並列性能に影響が考えられるが，2 軸並列化の効果で波数方向にも分割されているため，バンド方向の行列サイズによる，並列性能に影響が吸収されたためと考えられる．

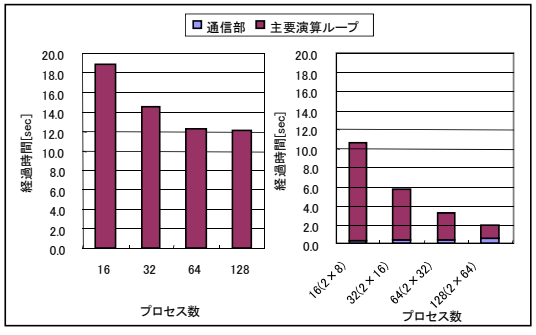

図 4 FX1 で測定した 2 軸並列化の並列性能の改善(区間 2)．左図は 1 軸並列時の並列性能，右図は 2 軸並列時の並列性能である．

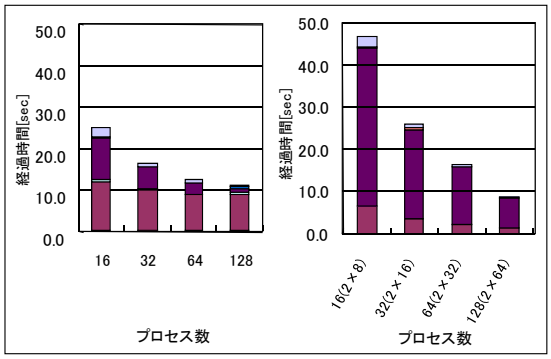

図 5 FX1 で測定した FFT を含むカーネルの並列特性(区間 8)．低並列時はオリジナルのプログラムと比較して遅くなっているが並列性能は良い．

3.3 FFT 高速化の評価結果

図 5は3.2節と同じ系を用いて FX1 で測定した V_{local} の逆 FFT を含むカーネル区間 8 の並列特性である．本測定は，超並列での評価はまだ出来ないため，比較的小規模での FFT 並列特性の評価結果である．2 軸並列化に伴い 16 並列の低並列時に性能が 24[s]→46[s]と劣化しているものの，128 並列では 11[s]→9[s]と並列性能は比較的良好である．低並列時の性能劣化は，2 軸並列化によりオリジナルの分割方法であるバンド方向の分割数が減少したことによる影響と考えられる．高並列時の並列性能の改善は，2 軸並列化に伴い並列粒度が増大したこと．更に 2 軸並列により全ノード間の全体通信が必要なくなったことで説明が可能である．

3.4 対角化計算並列化の評価結果

表 5は $HfSiO_2$1,536 原子アモルファス系を用いて「京」上にて測定した ScaLAPACK を用いた対角化を含む区間 9 の実行時間である．BLAS は並列性能が良いことが分かるが，対角化部分は既に並列性能が限界に達しており，並列数が増大するに従って性能が悪化している．これは，測定した原子数 1,500 の系では行列サイズ 5,000 元程度と小さいことが影響している．対角化すべき行列サイズがバンド数程度であるため，原子数の 2～4 倍程度であり，将来ターゲットとなる 10,000 原子程度の問題規模では 8 倍の 40,000 元程度の対角化が必要とされる．単純に考えると行列計算で演算量は 8^3~512 倍に増大し，並列数は 80,000 並列/2,000 並列～40 倍となるため，全実行時間は 1 桁増えることが推定できる．この行列サイズ規模の ScaLAPACK の並列特性と

表 5 ScaLAPACK を含むカーネルの並列特性.

カーネル	512 (16x32)	1024 (16x64)	1536 (16x96)	2048 (16x128)
区間9 (秒)	11.4	13.8	16.5	18.8
通信	0.0	0.0	0.0	0.0
BLAS	2.1	1.1	0.7	0.5
ScaLAPACK	8.8	12.0	15.1	17.5
他	0.5	0.7	0.7	0.8

して 1,000 並列程度の並列性能限界が知られている. しかし 1,000 並列程度まで並列性能が得られれば，アプリケーション全体に占める割合は 30%程度までには抑えられ, 影響は比較的小さいと予測できる.

3.5 まとめ

表 6に3.4節と同じ系を用いて,「京」2,048 並列で実測した実行時間と実行効率を記載した． BLAS 化した箇所は，40%以上の実行効率を達成し，SCF ループ全体を通じて 20%を越える実行効率を達成できた.

図 6は, 同じ系の各並列数でのカーネル種別毎の実行時間の内訳である．$HfSiO_2$ 1,536 原子の場合 2,000 並列程度まで並列性が保たれるのを確認した. この計算では, 既に原子数が並列数を下回っており, 今までの並列手法では分割粒度が細かく計算不可能な問題である.

仮にオリジナプログラムを用いても計算できたとして最適化前後の性能を比較すると, BLAS を導入されていない箇所の効率は 5%程度，FFT を含む箇所は，性能最適化前後で同程度の 5%以下の効率が想定されるので, 全体通信を含む本アプリケーションにて，実行効率で 20%を越えるのは画期的と言える.

表 6 「京」で測定した SCF ループ内の各カーネルのコスト一覧.

区間名	実行時間[sec]	浮動小数点演算ピーク比
SCF	39.78	20.11%
区間1(FFT)	0.02	2.19%
区間2(BLAS)	6.89	53.53%
区間3(FFT)	3.99	3.56%
区間4(BLAS)	1.88	64.76%
区間5BLAS,通信)	2.53	17.32%
区間6(FFT)	1.24	5.15%
区間8(FFT,BLAS)	4.16	16.56%
区間9(BLAS,ScaLAPACK)	12.31	3.88%
区間10(BLAS)	1.89	64.30%
区間11(FFT)	5.05	4.78%

4. 考察

PHASE の「京」向けの性能最適化について報告した．密度汎関数法の場合，演算密度が高いため，中心となるカーネルの大部分は演算順序などの工夫により BLAS Level3 の適用が可能となり，適用箇所の実行効率が数十%程度まで達成可能である.

また昨今のシステムの超並列化に伴い，FFT などの全体通信を含む通信アルゴリズムの性能低下が危惧されていたが，並列軸の追加により，通信をグループ化でき, 超並列にもある程度対応可能であることが示唆された．FFT に関しては，今後「京」では，多次元並列版 FFT が整備されるなど，更なる高速化が期待される．更に現在 FFT を行うためにパッキングなどの前処理を行っているが, この箇所のデータを連続化することで, 更に並列効率の向上が見込まれる.

対角化については，汎用ライブラリでは，それほど並列性能が見込まれないが, 必要とする計算が全空間での対角化ではなく, 部分空間という比較的小規模な対角化であるため，1,000 並列程度とある程度並列性能が保証されていれば, 全体の計算時間への影響は低く抑えられると考えている．今後，高並列版対角化ライブラリを導入することで, より高並列にも対応可能である.

本アプリケーションは「京」共用開始時に，SCF 収束回数を考慮しても, 理論上は 100,000 原子程度の電子状態計算が日単位で達成できる見込みである．しかし様々なデバイス特性を調べるために，分子動力学計算や反応中間体の計算などを実施する場合, 収束時間を短くしなければならないという要

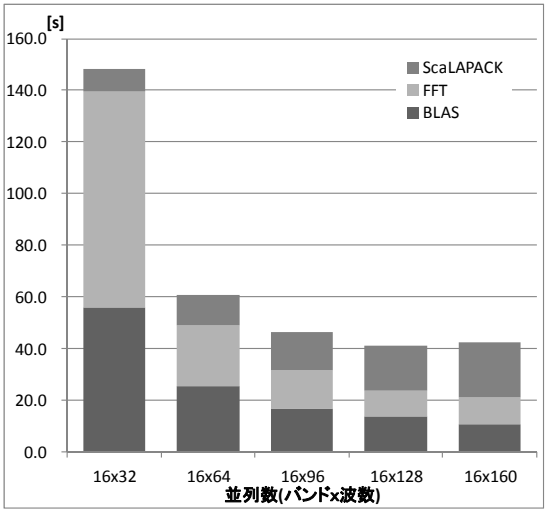

図 6 「京」で測定した各カーネルの並列特性

請から，10,000 原子程度の問題規模を全 80,000 プロセスを用いて計算する必要がある．この問題規模は，今までの手法では並列制約から実現不可能とされたが，今回の性能最適化により，初めてその目標達成の目処がたったといえる．

5. 謝辞

本性能最適化に際し御討論頂いた，アドバンストソフト株式会社の宇田毅氏，物質・材料研究機構の奈良純氏，富士通株式会社の上野潤一郎氏，井上晃氏，中島拓之氏，並びに理化学研究所次世代スーパーコンピュータ開発実施本部の諸氏に感謝します．本論文の結果は，理化学研究所計算科学研究機構が保有する京速コンピュータ「京」の試験利用によるものです．

参 考 文 献

[1] http://www.ciss.iis.u-tokyo.ac.jp/riss/project/device/

[2] P. Hohenberg and W. Kohn, "Inhomogeneous Electron Gas", Phys. Rev. B 136, pp864-871, 1964.

[3] W. Kohn and L. J. Sham, "Self-Consistent Equations Including Exchange and Correlation Effects", Phys. Rev. A 140, pp1133-1138, 1965.

[4] J. R. Chelikowsky and M. L. Cohen, "Nonlocal pseudopotential calculations for the electronic structure of eleven diamond and zinc-blende semiconductors", Phys. Rev. B 14, pp556-582, 1976.

[5] J. Ihm, A. Zunger and M. L. Cohen, "Momentum-space formalism for the total energy of solids", J. Phys. C: Solid State Phys., 12, pp4409-4421, 1979.

[6] S. G. Louie, K. Ho and M. L. Cohen, "Self-consistent mixed-basis approach to the electronic structure of solids", Phys. Rev. B 19, pp1774-1782, 1979.

[7] D. Vanderbilt, "Soft self-consistent pseudopotentials in a generalized eigenvalue formalism", Phys. Rev. B 41, pp7892-7895, 1990.

[8] K. Laasonen, A. Pasquarello, R. Car, C. Lee and D. Vanderbilt, "Car-Parrinello molecular dynamics with Vanderbilt ultrasoft pseudopotentials", Phys. Rev. B 47, pp10142-10153, 1993.

[9] R. M. Martin, “Electronic Structure: Basic Theory and Practical Methods”, Cambridge University Press, New York, 2004.

[10] L. J. Clarke, I. Stich and M. C. Payne, “Large-scale ab initio total energy calculations on parallel computers”, Comput. Phys. Commun., 72, pp14-28, 1992.

[11] A. Canning and D. Raczkowski, “Scaling first-principles plane-wave codes to thousands of processors”, Comput. Phys. Commun., 169, pp449-453, 2005.

[12] J. Hutter and A. Curioni, “Dual-level parallelism for ab initio molecular dynamics: Reaching teraflop performance with the CPMD code”, Parallel Computing, 31, pp1-17, 2005.

[13] F. Gygi, E. W. Draeger, M. Schulz, B. R. Supinski, J. A. Gunnels, V. Austel, J. C. Sexton, and F. Franchetti, “Large-Scale Electronic Structure Calculations of High-Z Metals on the BlueGene/L Platform”, ACM/IEEE SC’06, Tampa, USA , 2006.

[14] 今井 晴基, 森山 孝男, ”Blue Gene/P における第一原理分子動力学ソフトウェア PHASE の最適化”, 情報処理学会論文誌コンピューティングシステム (ACS), 2(2) , pp144-151, Jul. 2009.

[15] Maruyama, T., “SPARC64 VIIIfx: Fujitsu's New Generation Octo-core Processor for Peta Scale Computing.”, Hot Chips 21, 2009.

[16] Maruyama, T. “SPARC64 VIIIFX: A New-Generation Octocore Processor for Petascale Computing”, IEEE micro, 30 2, pp30-40, 2010.

[17] “SPARC International, The SPARC Architecture Manual (Version 9)”, Prentice-Hall, 1994.

[18] “Sparc Joint Programming Specification (JPS1): Commonality, architecture manual.”, Sun Microsystems and Fujitsu Ltd., 2002.

[19] “SPARC64VIIIfx Extensions”, Fujitsu Ltd., architecture manual, 2008.

[20] Ajima, Y., Sumimoto, S. and Shimizu, T., “Tofu: A 6D Mesh/Torus Interconnect for Exascale Computers.”, IEEE Computer, pp36-40, 2009.

[21] Toyoshima, T., “ICC: An interconnect controller for the Tofu interconnect architecture.”, Hot Chips 22, 2010.

[22] Yokozawa, T., Takahashi, D., Boku, T. and Sato, M., “Efficient parallel implementation of classical Gram-Schmidt orthogonalization using matrix multiplication”, Proceedings of Fourth International, Workshop on Parallel matrix Algorithms and Applications (PMAA’06), pp37-38, 2006.

[23] Hidemi Komatsu, Takahiro Yamasaki, Shinichi Ichikawa, “Massive Parallelization of First Principles Molecular Dynamics Code”, FSTJ Vol.44, No.4, pp449-457, Oct. 2008.

[24] Yamada, S., Imamura, T., Kano, T. and Machida M., “High-Performance Computing for Exact Numerical Approaches to Quantum Many-Body Problems on the Earth Simulator”, ACM/IEEE SC’06, Tampa, USA , 2006.

[25] Imamura, T., Yamada, S., and Machida, M., “Development of a high performance eigensolver on the petascale next generation supercomputer system.” Proceedings of Joint International Conference on Supercomputing in Nuclear Applications and Monte Carlo 2010 (SNA+MC2010), 2010.