

第一原理分子動力学・PHASEの超並列化 一次世代スーパーコンピューティングに向けて

Massive Parallelization of First Principles Molecular Dynamics Code

あらまし

PHASEは半導体・金属などの種々の物性を、量子力学に基づく電子状態計算によって説明・予測する第一原理分子動力学計算のシミュレーションコードである。このコードは文部科学省による革新的シミュレーションソフトウェア開発プロジェクトによって開発されたコードの一つで、この開発に富士通は参画している。

近年の計算規模の大規模化への要求により、数万原子規模の計算に対する需要が高まっており、このためには数万の演算器（CPU）を同時に動作させることによって高速に処理する、超並列処理が必要になる。そしてそのためには、実行するシミュレーションプログラムをそのような高並列に対応させるための超並列化が必要になる。

本稿では、このPHASEコードを超並列化するに当たって今回著者らの採った方法、すなわち大規模問題において演算負荷の高くなる部分として抽出したカーネル部分に対して、2次元分割の超並列化を実施した方法を説明する。そしてこの際に、CPU（プロセス）間のデータ転送量が演算量に比べて小さく取れ、予想として超並列化が高い性能を得ることを示す。

Abstract

PHASE is a first principles molecular dynamics simulation program for explaining and predicting various properties of semiconductors and metals through electron-state calculations based on quantum dynamics. Its code was developed by the Revolutionary Simulation Software project sponsored by the Ministry of Education, Culture, Sports, Science and Technology (MEXT), and Fujitsu has been participating in the development. As part of the recent trend toward large-scale computing, the need for simulating tens of thousands of atoms has been increasing, and to this end, there is a need for high-speed massively parallel processing on tens of thousands of central processing units (CPUs) simultaneously. This, in turn, requires that simulation programs be converted to massively parallel code to support such highly parallel operations. In this paper, we describe the method that we used to massively parallelize the PHASE code; it is a method for achieving massive parallelization of a two-dimensional decomposition of kernel sections having high computational load in large-scale problems. We show that the amount of data transferred between CPUs (processes) here is small compared with the computational complexity, indicating that the performance for massively parallel processing should be high.



小松秀実（こまつ ひでみ）
計算科学ソリューション統括部
所属
現在、スーパーコンピュータ向けプログラム高速化業務に従事。



山崎隆浩（やまさき たかひろ）
ナノテクノロジー研究センター
所属
現在、半導体材料の研究開発業務に従事。



市川真一（いちかわ しんいち）
計算科学ソリューション統括部
所属
現在、スーパーコンピュータ向けプログラムの性能評価業務に従事。

ま え が き

第一原理分子動力学計算とは、半導体・金属などの種々の物性を、量子力学に基づいてその電子状態を数値的に解くことによって予測・解明しようとする計算である。PHASE⁽¹⁾はそのための代表的シミュレーションコードで、文部科学省による革新的シミュレーションソフトウェア開発プロジェクトの中の1プロジェクトであるナノ・物質・材料・マルチスケール機能シミュレーション・システム開発プロジェクトによる開発コード⁽²⁾の一つである。富士通はこのプロジェクトに参画しており、著者の一人(山崎)はこのPHASEコードの開発者の一人である。

この第一原理分子動力学計算は、古典力学による近似を用いずに直接、量子力学に基づいた計算を行うため、扱う系の問題規模(原子数)が増えるにつれてその3乗で急激に演算量が増大し、現在要求されている数万原子規模の計算では、その実行に高速な演算能力を持つ計算機が必要となる。現状で通常行われている計算は、数百原子程度である。

一方で、現在構想されている高速科学計算機では、スカラCPUを数万個といった規模で多数連結し、各CPUに演算とデータを分担させることによって、全体として大規模問題を高速に処理する、超並列処理を目指している。そしてこのためには、計算するシミュレーションプログラム自体を超並列処理が可能な形に修正する超並列化が必要になる。

本稿では、まずPHASEプログラムの処理概要と現在の並列化方法およびその高並列実行における問題点を説明する。つぎに今回開発した、この問題を克服する超並列化方法を説明する。また、この超並列化を実施するに当たっては、PHASEプログラムの直接の超並列化を行う前に、まず大規模問題において特に演算負荷の大きくなる部分を抽出し、そこをモデル化した三つのカーネルコードを作成して超並列化を行ったので、その方法を説明する。また、PHASEの場合に演算負荷の大きい複数のカーネルコードを結合する際に必要となる、カーネルコード間のデータ並べ替え転送の方法を説明する。最後にプロセスあたりの転送量と演算量の比を評価し、プロセス間のデータ転送が演算量に比べて十分に小さく抑えられ、予想として高い並列スケーラビリティが得られることを示す。

PHASEの処理概要

PHASEの採用している擬ポテンシャル法は、原子核とコア電子をまとめて、凍結したイオン芯として扱い、価電子の状態だけを解く手法であり、材料特性の多くを予測することができる。擬ポテンシャルという名前は、価電子の波動関数が結合半径程度の距離より外側では真の波動関数に一致するようにしたまま、原子核近傍で大きな変動がないように調整したポテンシャルを使うことに由来する。原子核近傍の大きな変動を抑えることによって、波動関数 $\Psi(\vec{r})$ を平面波関数 $\exp(-i\vec{G}\cdot\vec{r})$ (\vec{G} は運動量空間の波数、 \vec{r} は実空間ベクトル)で展開しやすくなる(必要な \vec{G} の数が少なくてすむ)。擬ポテンシャルは、波動関数のs波、p波、d波^(注1)などの各成分に対応して異なる散乱能^(注2)を持つ局所項と、同じ散乱能を持つ局所項から成る。個々の価電子の波動関数を区別する添え字としてバンド(i)とk点がある。バンドは波動関数のエネルギー準位の違いを表し、k点は結晶が並進対称性^(注3)を持つことを反映した量子状態の指標で、バンドに分散を与える。この指標k点にプランク定数^(注4)を乗じて 2π で割ったものは結晶運動量と呼ばれる。半導体・絶縁体では電子の詰まっているバンドとそうでないバンドがk点によらず定まっていて、さらに大規模系ではバンド分散が小さくなるため、少数のサンプリングk点で十分によく電子状態を再現することができる。

PHASEのオリジナルコードの並列化は、バンドとサンプリングk点に関してなされているだけであり、半導体・絶縁体の大規模系ではk点方向の並列化の効果は限定的である。すると、この方法では数万CPU(プロセス)での並列化を行う際にバンド数が足りなくなり、十分な並列度を得ることができ

(注1) 個々の電子は原子核に近い内側から順に、s波、p波、d波と呼ばれる電子軌道に入る。

(注2) 原子核に近い内側のコア電子が、外側の価電子の軌道に与える影響のことである。

(注3) 結晶格子は格子定数の距離だけ格子を空間移動させても格子の状態が変わらない周期性を持っている。これを並進対称性と呼んでいる。これに対応して電子に保存する運動量が生じ、これがk点に対応した結晶運動量となる。結晶運動量はバンドのエネルギー準位に幅(分散)をもたらす。

(注4) 1個の電子の波動の特性を物理量に換算する、自然界の普遍定数。振動数にプランク定数を乗じるとエネルギーになり、波数にプランク定数を乗じると、運動量になる。

ない。また、仮にバンド方向のみに並列分割実行したとすると、プロセスあたりのバンド数が少なくなり、スカラ計算機上では演算性能が出にくい。さらに、波数方向に1次元並列を行う方法もあるが、この方法では総バンド要素数の自乗に比例するプロセス間転送が発生し、高並列時の転送負荷が大きい。そこで今回は、バンド (i) 方向と平面波基底の波数 (G) 方向を同時に並列化する、2次元並列化を行った。

PHASE高負荷部の抽出と超並列化

PHASEにおいて大規模問題を扱う際に最も演算量の大きくなる部分として、以下の三つがある。これを三つのカーネル部と呼ぶことにする。

- (1) 擬ポテンシャルと波動関数の積
- (2) Gram-Schmidt直交化
- (3) 3次元FFT

擬ポテンシャルと波動関数の積は、擬ポテンシャルの非局所項によって、個々の価電子がどのように影響されるかを計算する部分にあたる。擬ポテンシャルと波動関数の積を使って時間発展させた波動関数は、Gram-Schmidt直交化によって直交化条件を満たすようにする必要がある。そして3次元FFTは、波動関数を波数空間表示から実空間表示に、あるいはその逆の変換を行う。この変換は、局所ポテンシャルが電子に与える影響を計算する際や、価電子から電荷密度を構成する際に行う。これら三つの部分のオリジナルコードでの処理概要を図-1に示す。

今回は、これらのカーネル部をモデル化したコードを作成し、それに対して超並列化を行った。以下に、これらの超並列化方法について述べる。

- (1) 擬ポテンシャルと波動関数の積

PHASEでは各原子核に近いコアの電子は擬ポテンシャルとしてポテンシャル側に組み込み、その外側の価電子の波動関数のみを解く。このPHASEの高負荷部である非局所擬ポテンシャルと価電子波動関数の積を行う部分は、つぎのように表される。

$$|\Psi_i'\rangle = \sum_n D_n |\beta_n\rangle \langle \beta_n | \Psi_i\rangle$$

ここで、 $D_n |\beta_n\rangle \langle \beta_n |$ は、各原子の作る擬ポテンシャル、 Ψ は電子の波動関数、 n は擬ポテンシャルの軌道を識別する添え字と各原子を識別する添え字をまとめたもの、 i は電子の各波動関数（バン

ド）を区別する添え字である。この演算は、さらに以下のような前半部分と後半部分の計算に分かれており、原子数の3乗に比例する演算量を持っている。

擬ポテンシャル積・第1演算部：

$$\langle \beta_n | \Psi_i \rangle = \sum_G \beta_n^*(G) \Psi_i(G)$$

擬ポテンシャル積・第2演算部：

$$\Psi_i'(G) = \sum_n D_n \beta_n(G) \langle \beta_n | \Psi_i \rangle$$

ここで、 G は運動量空間の波数である。擬ポテンシャル β に関しては、参照エネルギー種別に関連してわずかな付随項が発生するが、ここでは上記のように省略し、簡略化している。

このようにモデル化した演算部に対してカーネルコードを作成し、それに対してバンド (i) 方向と波数 (G) 方向の2次元分割による並列化を行った。この際に、係数 D を β に繰り込んで上記の第1・第2の両演算部を行列積の形に帰着させ、さらにスカラ計算機上のキャッシュの再利用を促進するために、ループのブロック化を行った。この超並列コードの模式図を図-2に示す。2次元分割の並列化を行ったことによって、各方向のブロックサイズを十分に大きくとることが可能になり、演算効率を維持することができる。

なお、この並列化の際に、第1演算部と第2演算部の間で、波数方向のプロセス間の総和転送 (MPI_Allreduce転送) が新たに必要になる。

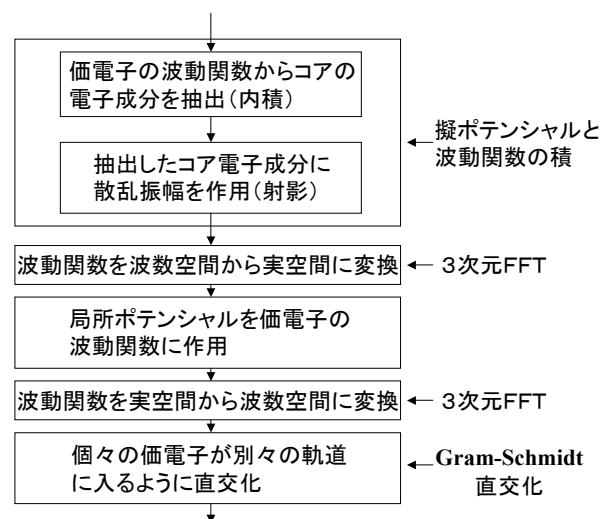


図-1 PHASEオリジナルコードの処理概要
Fig.1-Outline of original PHASE code.

(2) Gram-Schmidt直交化

ここは前記の擬ポテンシャル積部に続いて、原子数の3乗に比例する演算量を持っている。ここも擬ポテンシャル積部と同様に、前半部と後半部に分かれるが、この二つの処理は交互に繰り返される。すなわち、以下で*i*は1から*N* (全バンド数) まで繰り返される。

Gram-Schmidt直交化・第1演算部：

$$\langle \Psi_j' | \Psi_i \rangle = \sum_G \Psi_j'(G) \Psi_i(G)$$

Gram-Schmidt直交化・第2演算部：

$$|\Psi_i \rangle = |\Psi_i \rangle - |\Psi_j' \rangle \langle \Psi_j' | \Psi_i \rangle$$

G成分ごとに表示すると、

$$\Psi_i(G) = \Psi_i(G) - \sum_{j=1}^{i-1} \Psi_j'(G) \langle \Psi_j' | \Psi_i \rangle$$

このGram-Schmidt直交化部は、オリジナルのPHASEでは波数方向のみに並列化されており、ほかの部分の並列化方向 (バンド方向) と異なっているため、このGram-Schmidt直交化前後でプロセス間の担当データ並べ替え転送 (transpose転送) が発生する。しかし、今回著者らはこのtranspose転送を不要としかつ高並列度を得るため、擬ポテンシャル積などのほかの部分と同じ2次元データ分散配置をとって、波数 (*G*) とバンド (*i*) 方向の2次元の並列化を行った。この際に、上記の計算式における*i*を内側ループ、*j*を外側ループとし、前記の

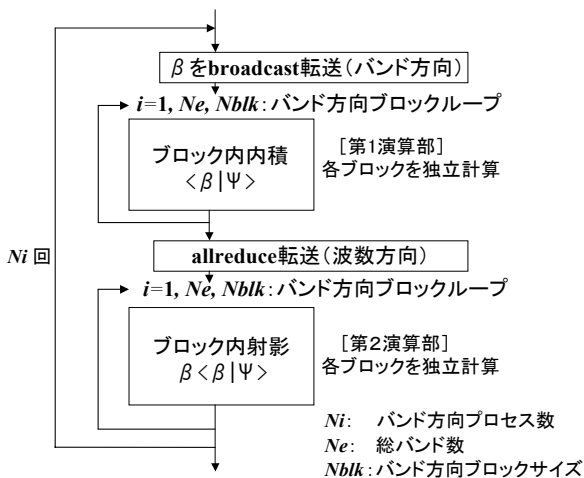


図-2 擬ポテンシャル積部・超並列コード模式図
Fig.2-Massively parallel code of pseudo potential product.

擬ポテンシャル積部と同様にループのブロック化を行った。

この超並列コードの模式図を図-3に示す。また、ここでの各プロセスへのデータ分散配置の様子を図-4に示す。ここでは各CPUの稼働率を向上させるため、バンド方向の1プロセスに複数のブロックを循環的に配置している。

(3) 3次元FFT

PHASEでは、局所ポテンシャルと電子波動関数の積を計算するなどの際に、3次元フーリエ変換 (FFT: Fast Fourier Transform) を用いて波動関数を波数空間から実空間に戻して計算する。したがって3次元フーリエ変換とその逆変換が何度か繰り返される。この部分は総波数を*N*とするとバンド数 × *N* log*N* に比例する演算量を必要とし、PHASEの3番目の高負荷部となっている。この部

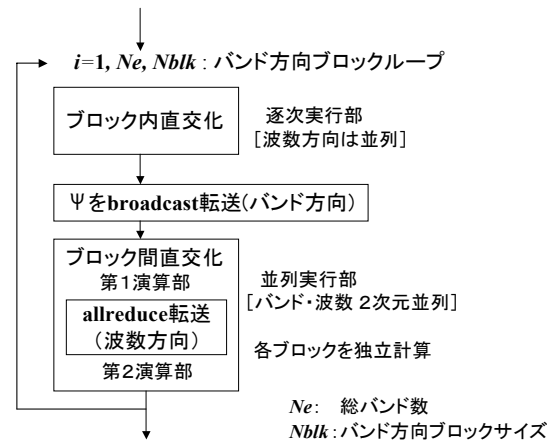


図-3 Gram-Schmidt直交化部・超並列コード模式図
Fig.3-Massively parallel code of Gram-Schmidt orthogonalization.

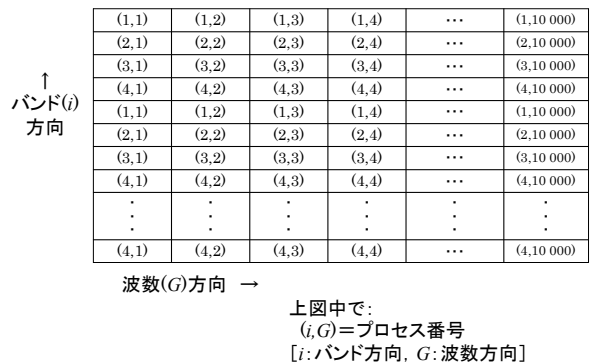


図-4 Gram-Schmidt直交化部・データ分散配置方法
Fig.4-Data distribution method in Gram-Schmidt orthogonalization.

分も前記の擬ポテンシャル積、Gram-Schmidt直交化の部分と同様に、バンド方向と波数あるいはバンド方向と実空間座標方向の両方について並列化を行うことができる。

今回著者らは、この3次元FFT部をバンドと x , y , z の波数方向 (G_x , G_y , G_z 方向) で並列化するに当たり、Eleftheriouの文献に示されている方法⁽³⁾と同様に波数方向内に対しても2次元分割の並列化を行った。すなわち、例えば x 方向の1次元FFTを行っている際には、波数の y 方向と z 方向について2次元の並列化を行うことになる。

ここで、 x 方向の1次元FFTから y 方向の1次元FFTに移る際には、各プロセス上の x 方向データと y 方向データの入換え転送 (y 方向のtranspose転送) が必要になる。同様に、 y 方向の1次元FFTから z 方向の1次元FFTに移る際には、各プロセス上の y 方向データと z 方向データの入換え (z 方向のtranspose転送) が必要になる。

カーネル結合転送部

前章でPHASEの三つのカーネル部の超並列化の方法を示したが、このPHASEの三つのカーネル部では、それぞれデータ (波動関数) の並びが異なっている。したがって、これら三つのカーネル部間でデータを並べ替えるプロセス間データ転送が必要になる。以下に、超並列時における波数方向、バンド方向それぞれの転送方法を述べる。

(1) 波数方向並べ替え転送

PHASEでは3次元FFTに関連する部分を除いて、波数の絶対値に関する上限値 (カットオフ) を設定しており、したがって波数の絶対値順にデータを並べ替えて計算を行っている。具体的には、擬ポテンシャル積部とGram-Schmidt直交化部では波動関数データ $\Psi_i(G)$ は波数の絶対値 ($G^2 = G_x^2 + G_y^2 + G_z^2$) の順に並べられているが、3次元FFT部では x , y , z 順 (G_x , G_y , G_z 順) に並べられている。したがって、3次元FFT部の前後で波数方向のデータ並べ替え転送が必要になる。ここで超並列時には、各プロセスの担当する波数の要素数よりも、波数方向のプロセス数の方がはるかに大きくなるため、波数の1要素ずつをインデックスに従って目的のプロセスに直接転送することになる。

(2) バンド方向並べ替え転送

PHASEではGram-Schmidt直交化に入る前に、直交化すべき波動関数をいったんそのバンドに対応した固有値 (固有エネルギー) 順に並べ替えることによって、収束を早めている。したがって、ここでバンド方向のデータ並べ替え転送が必要になる。今回の超並列化ではバンド方向に割り当てられるプロセス数は、各プロセスが搭載しているバンドの要素数に比べて少ないため、各プロセスに転送するデータをあらかじめまとめて、プロセスごとに1度に転送する方式を採用した。この転送方式の模式図を図-5に示す。

また、数万個の波動関数を固有値順に並べ替えるためには、あらかじめ固有値をその大きさ順 (小さい順) に並べ替えるソート処理によって、インデックスを作成しておくが必要になる。今回はこのソート処理を並列で行うために、並列実行に適した分布数え上げソートと呼ばれる方法と単純挿入法とを組み合わせた方法を採用した。すなわち、各プロセスへのデータ分散を分布数え上げソートにより行い、各プロセスに振り分けられたデータのプロセス内でのソートは単純挿入法によって行い、最後にこれを融合した。

超並列時の演算量と転送量

今回、超並列化を行ったPHASEカーネル部で、演算量と転送量ともに最も負荷が高い部分は、擬ポテンシャル積部とGram-Schmidt直交化部である。

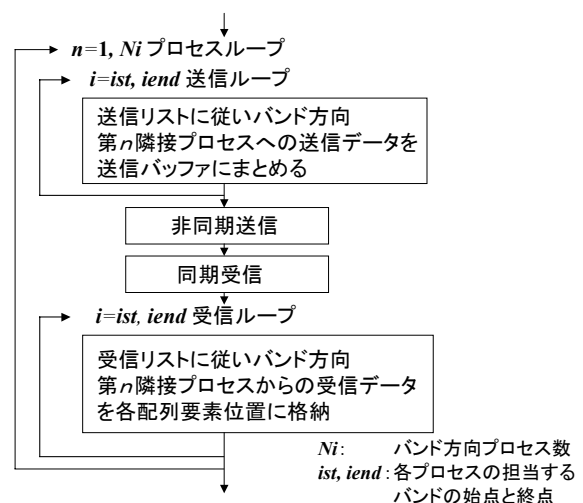


図-5 バンド方向並べ替え転送コード模式図
Fig.5-Data transfer code in band direction.

これらの部分と3次元FFT部の総演算量と総転送量は、 N_e を総バンド数、 N_f を総波数、 N_i をバンド方向プロセス数、 N_k を波数方向プロセス数とすると、プロセスあたりで近似的に以下ようになる。ここで、擬ポテンシャルの総数をほぼ総バンド数 N_e に等しいと近似した。

(1) 擬ポテンシャル積と波動関数の積

演算量： $16 \times (N_e)^2 \times N_f / (N_i \times N_k)$

転送量： $16 \times (N_e)^2 / N_i$ (バイト)

[Allreduce転送]

(2) Gram-Schmidt直交化

演算量： $8 \times (N_e)^2 \times N_f / (N_i \times N_k)$

転送量： $8 \times (N_e)^2 / N_i$ (バイト)

[Allreduce転送]

(3) 3次元FFT

演算量： $5 \times N_e \times N_f \times \log N_f / (N_i \times N_k)$

転送量： $16 \times N_e \times N_f / (N_i \times N_k)$ (バイト)

[transpose転送]

ここで、1万原子規模での各パラメータ値は、 $N_e = 50\,000$ 、 $N_f = 1\,000\,000$ (想定プロセス数 $N_i = 4$ 、 $N_k = 10\,000$) 程度の大きさになる。Gram-Schmidt直交化において、前処理として1ブロック内を直交化する処理があり、ここではバンド方向には並列実行されない。したがって、各CPUの稼働率を向上させるには、波数方向のプロセス数の比率を大きくすることが望ましい。これは言葉を変えると、1CPU (1プロセス) の担当するバンドブロック数を大きく取り、1ブロック分の前処理を目立たなくすることに相当する。したがって、ここではバンド方向プロセス数 N_i を少なく取っている。この前処理部のプロセスあたりの演算量は $8 \times N_{blk} \times N_f \times N_e / N_k$ 、転送量は $8 \times (N_{blk} - 1) \times N_e$ である。ここで N_{blk} はブロックサイズであり、1プロセスに複数のバンドブロックを置くことによって、その後の処理に比べてこの前処理部の演算量と転送量を小さく抑えることができる。

上記において、擬ポテンシャル積部とGram-Schmidt直交化部では転送量と演算量の比が

$$(\text{転送量}) / (\text{演算量}) = N_k / N_f$$

となり、プロセスあたりの波数要素数の逆数となっている。通常、PHASEでは波数要素数は原子数の数百倍程度取られるため、想定プロセス数が $N_i = 4$ 、 $N_k = 10\,000$ 程度の場合には、上記の転送量・演算

量比は十分に小さくなり (~1/100程度) 転送時間を演算時間に比べて短く抑えることができる。一方で、3次元FFT部については、演算量・転送量ともに前の2部分に比べてかなり小さくなっている。

また、上記の三つのカーネル部を結合する二つの結合転送部については、ともに総転送量は

$$16 \times N_e \times N_f / (N_i \times N_k) \text{ (バイト)}$$

となり、上記の擬ポテンシャル積部とGram-Schmidt直交化部の総転送量に比べて十分小さくなっており (~1/500程度)、トータルの演算性能の大きな低下はもたらさないと考えられる。

仮に、転送量の最も大きな擬ポテンシャル積部とGram-Schmidt直交化部のMPI_Allreduce転送についてRecursive Halving法⁴⁾を採用し、データ転送能力として一様に4 Gバイト/秒を仮定し、演算性能としてプロセスあたり100 Gflops (100 G演算/秒) を仮定すると、転送時間と演算時間の比は次のようになる。

$$(\text{転送時間} / \text{演算時間}) = 2 \times (N_k / N_f) \times (100 \text{ G} / 4 \text{ G})$$

この結果から、2次元分割によって転送時間は演算時間に比べて小さく抑えられることが分かる。

波数方向のみの1次元分割を行った場合の、各演算部と通信部の実行時間の内訳の予想比率を図-6に示す。また、波数方向のみの1次元分割の並列化を行った場合と、今回の2次元分割の並列化を行った場合の、8プロセスでの実行時間に対する各プロセス実行数での高速化倍率の予想グラフを図-7に示す。最も大きな転送負荷となるAllreduce転送について、2次元並列時には(1/バンド並列数)に転送量が削減されるため、1次元並列に比べて転送効率が良い。このため、とくに数万プロセスでの並列実行の際に、

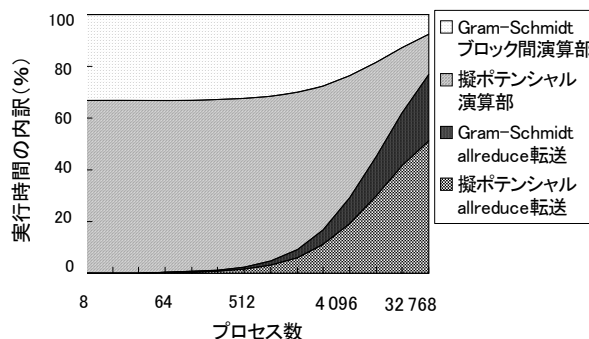


図-6 1次元波数並列化による実行時間内訳の推定
Fig.6-Breakdown of estimated execution time by one-dimensional decomposition in wavenumber direction.

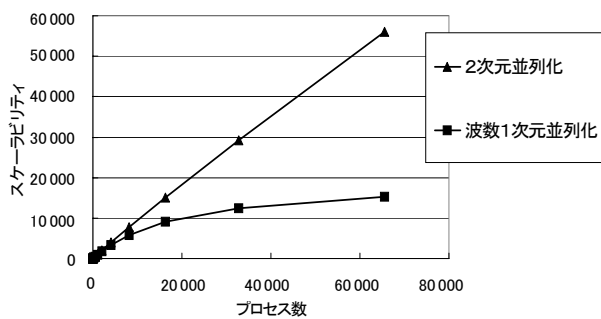


図-7 波数とバンドの2次元並列化によるスケラビリティの推定

Fig.7-Estimated scalability by two-dimensional decomposition in wavenumber and band directions.

1次元並列に比べて2次元並列が非常に良い性能を得られることが、グラフから分かる。

む す び

今回、著者らは第一原理分子動力学コードPHASEの数万CPUに対する超並列化の可能性を示すために、大規模問題の計算時での三つの高負荷部を抽出し、それらをモデル化したカーネルコードを作成し、このカーネルコードに対して超並列化を行った。そして、これら三つのカーネルコードを結合するために必要な、データ並べ替え転送部を記述し、最終的に3カーネルの結合コードを作成した。

さらに、このカーネル結合コードの各部の演算量と転送量を調べ、数万CPUでの並列実行時に演算量に比べてCPU間のデータ転送量が小さく抑えられることを見だし、数万プロセス（CPU）での超並列化が可能であることを示した。

今後は、ここで作成した結合カーネルコードを用いて、実機による性能評価を行う予定である。本稿では大規模並列時における見積もりで、低並列時と同じ構成のネットワークを仮定したが、実アプリケーションでよく使用されるMPI_Allreduce転送のような集団通信に対して、高速な処理を行う大規模ネットワークの実現が期待される。

なお、今回の研究は文部科学省によるプロジェクト「ペタスケール・システムインターコネクタ (PSI) 技術の開発」のもとで行われた。

参 考 文 献

- (1) RSS21 文部科学省次世代IT基盤構築のための研究開発プログラム 戦略的シミュレーションソフトウェアの研究開発：PHASEの詳細情報。
<http://www.ciss.iis.u-tokyo.ac.jp/rss21/theme/multi/material/index.html>
- (2) RSS21 文部科学省次世代IT基盤構築のための研究開発プログラム 戦略的シミュレーションソフトウェアの研究開発：ソフトウェアの公開。
<http://www.ciss.iis.u-tokyo.ac.jp/rss21/result/download/index.php>
- (3) M. Eleftheriou et al. : Scalable framework for 3D FFTs on the Blue Gene/L supercomputer: Implementation and early performance measurements. *IBM J. RES. & DEV.* Vol.49, No.2/3, p.457-464 (2005).
- (4) R. Thakur et al. : Optimization of Collective Communication Operations in MPICH. *Int. J. HPC. Appli.* Vol.19, No.1, p.49-66 (2005).