

大規模シミュレーション用並列ソフトウェアの 高速最適化技術研究

Research of Parallel Software Optimization for Large-scale Simulation

(財) 高度情報科学技術研究機構
計算科学技術部
飯塚 幹夫、南 一生、中村 壽

近年の地球シミュレータなどの高性能スーパーコンピュータの発達により、高速演算による大規模シミュレーションが可能になり、計算科学を利用した新発見や新技術開発への挑戦が始まっている。一方、将来へ向かう次世代スーパーコンピュータ開発では、プロセッサはマルチコア等に複雑化し、しかもその数は数十万にも及ぶものと予想される。このため、膨大な並列処理が要求され、従来のような応用ソフトウェアの並列化、演算性能向上技術では、対応が相当に難しくなるものと予想される。そのため、次世代のペタフロップス級スーパーコンピュータの効果的な利用と新しい発見や新技術を開発しやすくするために、応用ソフトウェアの新しい高速最適化技術が課題となっている。本稿では、財団で現在取り組んでいる高速最適化技術の研究開発について紹介する。

1. まえがき

計算科学の推進を目的として、先端的な計算資源を利用する大規模シミュレーションに関わる幅広い高度ソフトウェア技術の研究を進めている。近年、単体プロセッサの性能向上の限界からスーパーコンピュータのプロセッサ数は急増し、かつプロセッサもマルチコア化が進み、応用ソフトウェアの超並列計算機への対応が課題となっている。またメモリの性能向上のスピードは遅くメモリ・ウォール問題は深刻化する一方であり、応用ソフトウェアの性能向上のためますます高度な技術が要求されるようになっている。

このような背景のもと、次世代大規模計算資源を利用したペタフロップス級ハイエンドコンピューティングの効果的な利用と高度な計算科学技術の発展のために、高速最適化技術の研究開発に取り組んでいる。大規模並列科学技術計算コードは10万行を超える大規模

複雑なものもあり、次世代大規模計算資源の利用において、これまでのようにコード局部をもぐら叩きのチューニングする方法では局部的な最適化は達成できても、十分な高速化の達成は困難である。本稿では、このようなこれまでのコードチューニングの問題を克服することを目的に進めているコード全体を大局的に俯瞰し物理的、数学的、アルゴリズム的、体系的にアプローチする高速化技術とその体系化の研究開発について紹介する。

2. 大規模シミュレーションのための高速最適化技術の重要性と課題

計算科学 (Computational Science) は、数学的モデルとその数値的解法を構築し、計算機を駆使して科学技術上の問題を解決するものである。計算科学は科学の第3の形態と見なされ実験・観測と理論の間を補間するものと考えられている。計算科学への期待は高

いものの、計算資源の制限から1990年代後半まで、計算科学は実験・観測と理論の補助的手段であるとの考えが一般的であった。一方1990年代に航空宇宙研究所の数値風洞（166プロセッサ、理論ピーク性能0.28Tflops）、米国ASCIプロジェクト（1995年に開始、2004年に理論ピーク性能100Tflopsを目指した）等の並列スーパーコンピュータの開発が進んだ。2002年には地球シミュレータが登場し、その高い実効性能により応用ソフトウェアでの安定したテラフロップス級の大規模シミュレーションが可能となった。地球シミュレータ上では、地球科学の応用ソフトウェアだけでなく、ナノ（カーボンナノチューブ）・バイオ系、物理系（高温超伝導体を用いたテラヘルツ発振）、工学系の応用ソフトウェアも高性能で実行可能であることが分かった。そのため、大規模計算資源を利用した大規模シミュレーションによる新発見や新技術開発への期待が高まった。

計算科学技術の発展に寄与しているのはスーパーコンピュータの性能向上である。CRAY-2（1985）とBlue Gene/L（2005）を比較すると、この20年間で性能向上は理論ピーク性能で約190万倍にも達する（図-1）。

我が国では、次世代スーパーコンピュータの開発は5つの国家基幹技術の1つとして取り上げられ、2012年には10Pflops級のスーパーコンピュータの稼動が予定されている。このようなスーパーコンピュータの性能を効果的に使い、高い実効性能で応用ソフトウェアを実行し、逸早く新発見に繋がるデータを得ること、また新技術開発のための問題解決の解へ短時間で到達することが、今後の我が国の科学技術で優位性を保つために有効である。そのため、大規模シミュレーションでの高速性の重要性は明らかである。

一方、ムーアの法則の限界から単一プロセッサの性能は頭打ちとなり、スーパーコンピュータのプロセッサ数はますます増加し、消費電力の問題からプロセッサはマルチコア、メニーコアとなる傾向である。またメモリ性能の向上は遅く、メモリの階層構造はますます複雑になっていく（共有や部分共有キャッシュ等）傾向である。このような増え続ける並列性と悪化するメモリ・ウォール問題は、応用ソフトウェアの高速化を以前にも増して難しくしている。従って、増え続ける

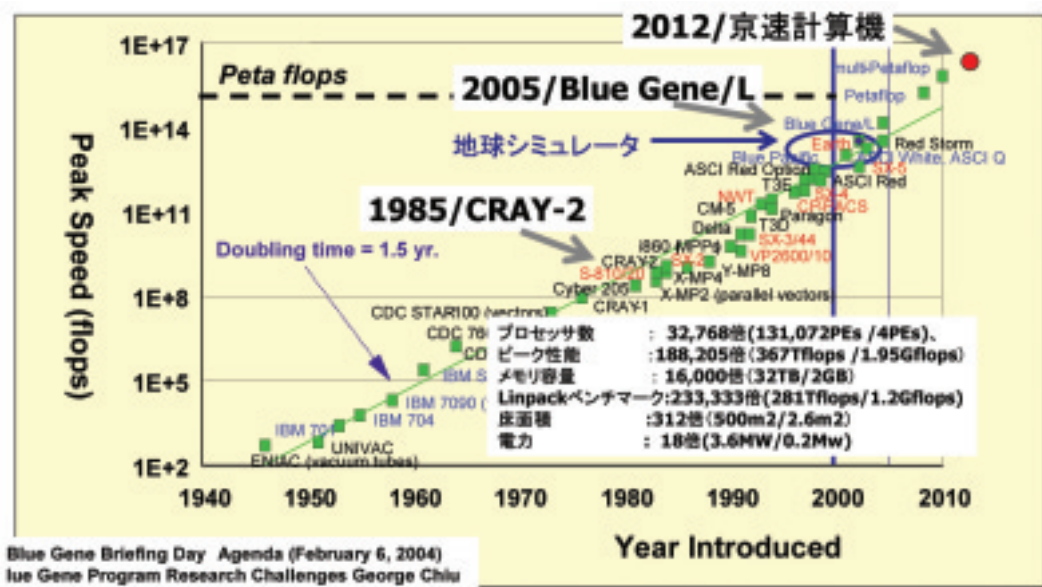


図-1 スーパーコンピュータの20年間の性能向上 (CRAY-2とBlue Gene/Lの比較)

並列性の要求に応える超並列計算技術とメモリ・ウォール問題を克服する最適なデータ配置技術は、大規模シミュレーションにおける高速最適化の大きな課題となっている。また、図-2に示すように、高速化での障害となるアルゴリズム上のボトルネックの発見、分析、性能高速化指針作成に有効な、高性能計算機を対象とした演算性能推定技術は、高速計算機の効率的な利用、さらにはその設計、開発にも寄与する重要なものである。今後、ますますアーキテクチャが複雑化していく傾向にある高性能計算機に対して、こうした高度な性能推定法を開発していくことは高速最適化技術のもうひとつの重要な課題である。

以上のような背景の下、我々が進めている高速最適化技術の研究開発について、以下に紹介する。

3. 大規模シミュレーションのための高速最適化技術の研究開発

3.1 これまでの高速最適化技術の課題

これまでの高速化技術は、例えば図-3に概略的に示すように、計算機アーキテクチャの発展とともにソフトウェアシステムエンジニアや多くのユーザによって、長年の経験を基に蓄積されたものであり、それなりに効果をもたらして来た。しかし、その現状を仔細に見れば、単体性能の向上であれ、並列性能の向上であれ、サブルーチンやループでの最

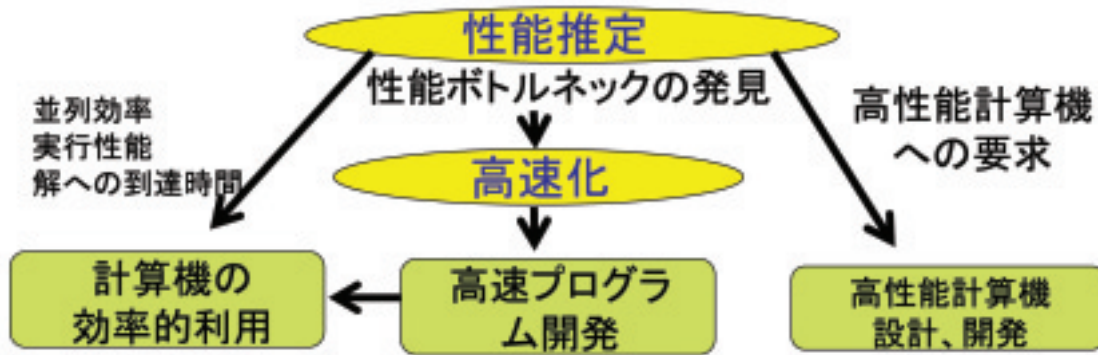


図-2 高速最適化技術の効果

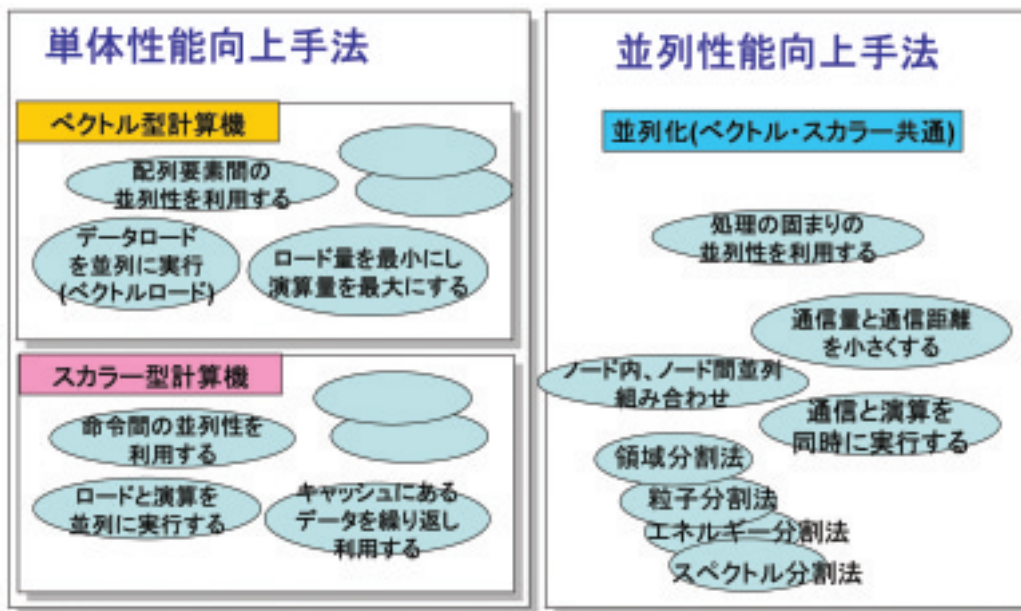


図-3 これまでの高速化技術の分類

適化である。つまり、プログラムレベルの局所的な並列性とデータの局所性を利用する部分的な最適化であることがわかる。また、並列性能の向上においては、物理、工学分野のシミュレーション研究者により大局的に並列性のある並列手法が、例えば領域分割法、粒子分割法、エネルギー分割法等が、発見、開発された。しかし、これらの手法の並列性を複数組み合わせ、かつソフトウェアの全体から部分的な階層でのいろいろな並列性との組み合わせによる高速化、全体性能面から見たバランスによるソフトウェア全体での高速最適化等での高度化は十分ではなかった。また、スカラー型計算機向き、ベクトル型計算機向きで異なる最適化の方法を採ってきており全体での統一性の追及も不足していた。

このように、これまでの高速化技術は、個々の技術は優れたものであったが、局所的で個別的という側面が強かった。従って、これまで長年をかけて蓄積された高速化技術・知識は、十分に生かされず、断片的個別的な高速化技術の活用で、コードレベルでもぐら叩きのみに最適化が行なわれることが多かった。

しかし、今後登場すると予想される数万～数百万のマルチコア、メニーコアプロセッサからなるペタスケール、エクサスケールのスーパーコンピュータが要求する超並列性、メモリ・ウォール問題に対応するには、これまでの高速最適化技術では不十分である。

3.2 本研究での高速最適化技術の研究開発

本研究では、応用ソフトウェアに関し、その出発点となる物理・数学モデル、またその解析手法、数値アルゴリズム、さらにプログラム表現、データ特性などまでを対象として、利用する計算機のアーキテクチャと対比させつつ、局所的かつ全体的、または階層的に数値処理プロセスを仔細に分析し、そこに顕在、潜在する複数の並列性、データの局所性を見付だし、これらを効果的に組み合わせて

利用し、ソフトウェア全体の性能を高めていくという観点に立って、統合的な次世代向けの高速最適化技術の研究開発を進めている。また、これと相補的な関係にある性能に関する定量的な評価技術の研究開発も進めている。

(1) 統合的な高速化技術

応用ソフトウェアの高速化とは、簡単に言えば、其々のプロセッサの休止状態を極力少なくなるように、かつできるだけ多くのプロセッサを使えるようにし、ソフトウェアの演算性能を向上させ、結果として利用する計算機資源の経済効果を高めることである。このために、対象とするソフトウェアのプログラムを改良または再構築、さらには性能効率の高いソフトウェア設計指針を作り出すことである。このように目標は単純明快であるが、複雑化するスーパーコンピュータと大規模な計算科学技術用応用ソフトウェアを対象に、具体的に性能をどのように高めるかは至極難しい問題である。

高速化の基本は、スカラー型、ベクトル型計算機であれ、まずプロセッサに十分なデータを供給し、プロセッサの作動に休止状態を作らないこと。また、並列する数多くのプロセッサを通信させながら、同時並行的に計算処理させて、与えられたプロセッサの能力を極力使い切ることである。そのためには、プロセッサ数が数千以下の比較的、小、中規模の並列処理の場合には、まず既存コードの顕在する並列性、データの局所性を最大限に利用することが目標とされる。例えば、ループ添字について依存性がなく十分な並列性があれば、ベクトル型計算機ではベクトル処理が有効になり、スカラー型計算機ではループ内の演算について十分な並列性があれば、命令レベルの並列処理により、高速化が図れる。また、スカラー型、ベクトル型計算機いずれでも、ロード・ストアと演算の比を改善させ、データの局所性を高めることで、結果的に高

速化に繋がる。さらに、スカラー型計算機では、データのキャッシュブロック化によりデータの局所性を向上させ高速化が図れる。

しかし、今後、出現するマルチコアやメニーコアなどプロセッサ数が数万から十万程度に及ぶ次世代型スーパーコンピュータを考えた場合、高速化においては、従来型の並列性の利用、データの局所性の改良のみでは不十分になると予想される。

プロセッサ数が増加し超並列になると、プロセッサ間の通信量が増大し、その負荷が演算性能を減じ、また小さい非並列部もその影を増し演算性能を減じるため、大規模な並列性への配慮が非常に重要となる。このため、超並列において、高い演算性能を達成するためには、対象となるアプリケーションの顕在する並列性は勿論の事、これまでに気づかれていない潜在的な並列性などを明らかにし、データの局所性と効果的に組み合わせるなど統合的な並列処理スキームを考えていく必要がある。

具体的には、プログラム全体にわたる既存の並列性の効果的な利用に加え、潜在的な並列性の発見、非並列部（逐次処理）の回避と低減、さらにデータの局所性の発見と利用が、枢要である。このため、ソフトウェアに

関する総合的分析の視点として、背景となる物理・数学モデル、またそれらに対応する数理解手法とアルゴリズム、加えてプログラム表現などを取り入れた。また、根本的に性能を向上させるため、物理・数学モデル、離散化モデルのレベルで大局的にコードを改造することを視野に、アプリケーションをそれらの観点も含めて計算カーネルへ分解し、統合する分析手法を導入した（図-4）。このように、本研究では、ソフトウェアの背景にある物理・数学モデル、解析手法にまで立ち返る方法を取り入れた。

また、物理・数学モデルから計算機の各レベルにおいて、複数の並列性と局所性がある（図-5）。これらを総合的に分析し組み合わせて統合的に利用することでアプリケーションの並列度を大幅に高め、またデータの局所性を大幅に改善できると考えた。

さらに、本研究では、これまでに蓄積された計算科学技術の知識をさらに発展させるため、従来の技術を、並列性の発見と統合利用、データの局所性の発見とその統合利用という観点から再構成することが有効であろうと考えた（図-6）。このような単純な視点の導入で、アプリケーションの分析と発見さ

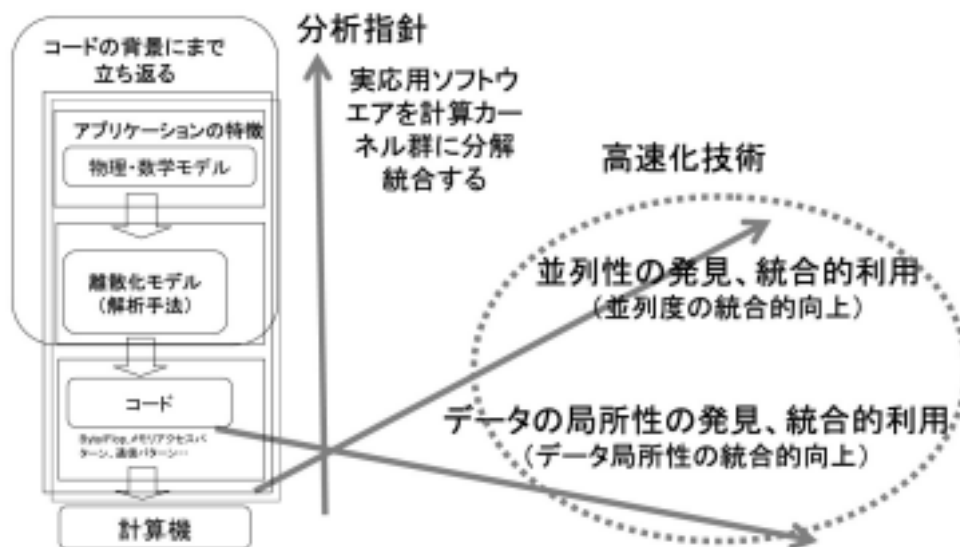


図-4 統合的高速化技術-1

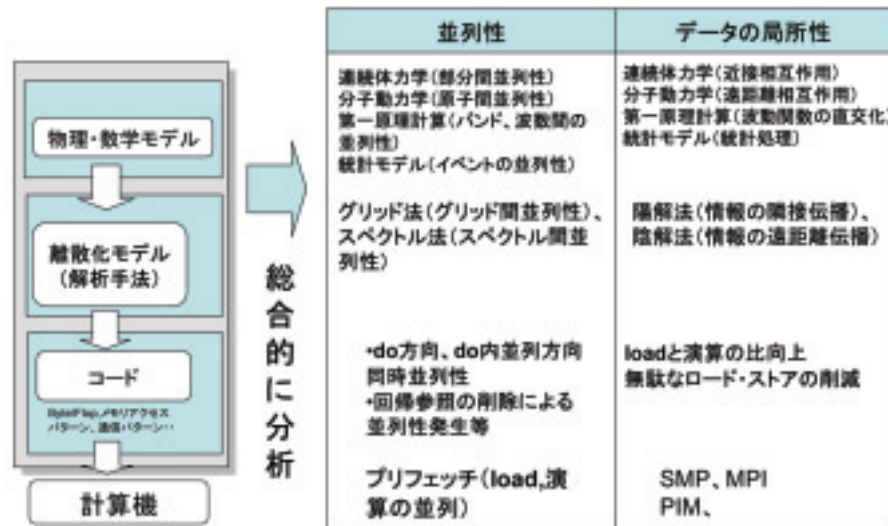


図-5 統合的高速化技術-2

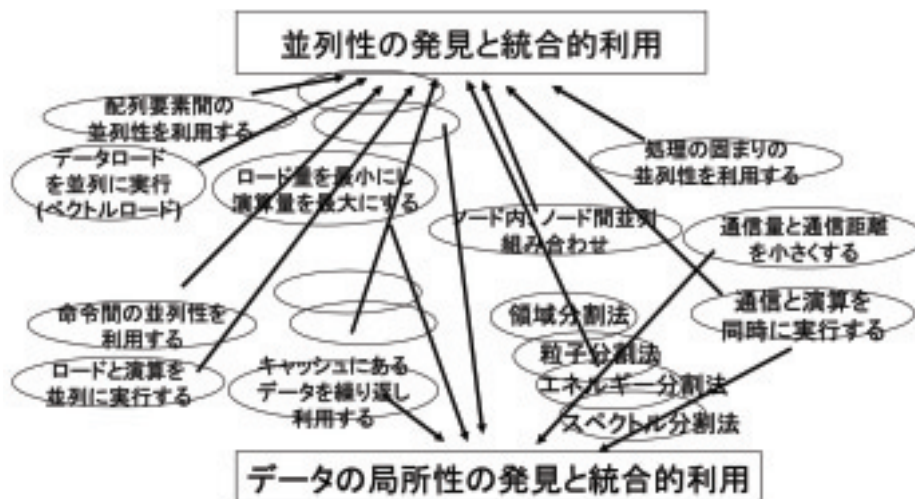


図-6 統合的高速化技術-3

れた並列性、データの局所性の統合的利用が容易となると期待される。

(2) 性能推定法の開発

科学技術分野の応用ソフトウェアの並列効率、演算性能の評価と推定は、従来、ソフトウェアの複雑さと大規模計算の実施に経験が必要であることから、通常、比較的小規模な問題、少数プロセッサで全体を丸ごと測定した限定的な性能データに基づく実験近似式により行なわれてきた。しかし、一般に科学技術応用ソフトウェアは、計算負荷の集中する多数の計算処理ブロック(計算カーネル)か

ら構成され、問題規模と利用プロセッサ数またそれらの変化量に依存して、並列効率、演算性能は変化する。従って、従来のような限定的性能データに基づく方法のみでは、数万から数十万プロセッサなどの次世代型スーパーコンピュータでの大規模シミュレーションの性能を推定することは困難である。

本研究においては、プログラム全体を計算カーネル群に分解し、計算カーネル毎の並列性能、演算性能をモデリングし、それらを統合し、ソフトウェア全体の性能推定を行う方法を採用した。具体的には、まず対象となるソフトウェアを、物理、数学モデル、数値解

析法とその処理フローの観点からおおまかなブロックに分割する。次に、計算負荷の分布測定を行い、負荷の高いブロックをさらに物理数学モデル、数値解析法とその処理フローの観点からさらに小さなブロックに分割していく。その小ブロックの中から、計算負荷の高い順に計算カーネル群を抽出する。計算カーネルの並列化率、通信オーバーヘッド、単体性能を計測し、物理数学モデル、数値解析法の知識を援用しモデル化する。そして、システム全体性能はこれら計算カーネル群全体の総合的性能としてモデル化する。性能モデル式は、並列オーバーヘッドの影響を加えてアムダールの法則を拡張し、計算カーネル群からなる応用ソフトウェアの全体性能に対し、解への

到達時間 (time to solution)、加速率、並列化率や実効効率について以下のように作成した。

つぎに、具体的な研究成果について説明する。

4. 高速最適化技術の適用例

文部科学省次世代IT基盤構築のための研究開発「戦略的革新シミュレーションソフトウェアの研究開発」(平成17年から19年)において開発された、代表的な2つのコード、マルチフィジックス流体コード、Frontflow/Blue (FFB:有限要素法)とナノ物質・材料シミュレーションコードPHASE(密度汎関数法による平面波モデル)を対象に本高速最適化技術の適用例を紹介する。

●time to solution

$$t^n = \sum_k t_k^n = \sum_k t_k^1 Q_k^n = t_{com}^n + t_{cal}^n = \sum_k V_k^1 / f_{e,k}^1 Q_k^n$$

$$t^1 = \sum_k t_k^1 = \sum_k V_k^1 / f_{e,k}^1$$

t^n : プロセッサ数 n でのシステム全体の処理時間

t_{com}^n : プロセッサ数 n でのシステム全体の通信時間

t_{cal}^n : プロセッサ数 n でのシステム全体の計算時間

V_k^1 : プロセッサ数 1 での計算カーネル k の計算時間

$f_{e,k}^1$: プロセッサ数 1 での計算カーネル k の実効性能

●加速率

$$Q_k^n(V_k^1) = 1 + \beta_k^n - (1 - 1/n)\alpha_k^n(V_k^1)$$

β_k^n : プロセッサ数 n で計算カーネル k の並列オーバーヘッド

α_k^n : プロセッサ数 n で計算カーネル k の並列化率

●並列化率

$$\frac{(1 + \beta)t^1(V_{cal}^1)}{t^n(V_{cal}^n)} = \frac{\sum_k V_k^1 / f_{e,k}^1}{\sum_k V_k^1 / f_{e,k}^1 Q_k^n} = \frac{1}{1 - (1 - 1/n)\hat{\alpha}^n(V_{cal}^n)}$$

$$\hat{\alpha}^n(V_{cal}) = (1 - 1/n)^{-1} \left(1 - \frac{t^n}{t^1(1 + \beta)} \right)$$

●実効効率

$$f_e^n / f_{th} = V_{cal} / t^n = (\sum_k V_k^1) / (f_{th} \sum_k t_k^n) = (\sum_k V_k^1) / (f_{th} \sum_k V_k^1 / f_{e,k}^1 Q_k^n)$$

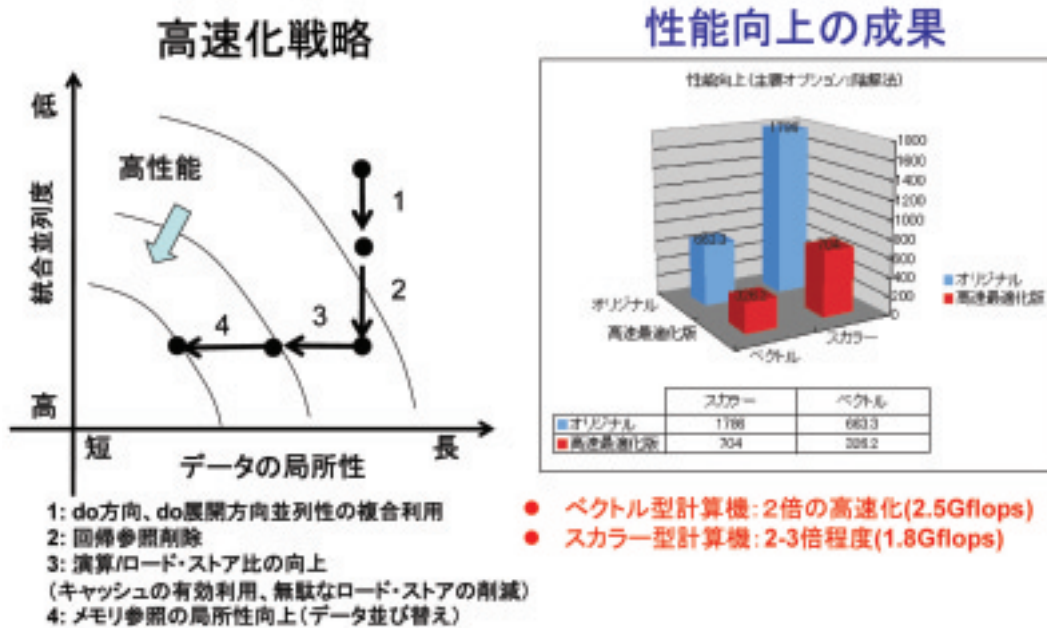


図-7 FFBに対する統合的高速化技術の応用例

(1) 統合的高速化の結果例

FFB: 従来、高速最適化プログラミングはスカラー計算機とベクトル計算機でそれぞれ異なるものであった。しかし、本研究では、ループに内在する独立変数、命令など多重の並列性を統合的に利用し、スカラー計算機とベクトル計算機について、完全に同一のプログラミングでありながら、両型の計算機で共に2~3倍以上の高速化を達成することができる技術を開発した(図-7)。

並列性の利用、データの局所性の向上は、スカラー型計算機であれ、ベクトル型計算機であれ高速化の要である。具体的には、Doループの添字変数の回帰参照の削除による並列性の確保及びループアンローリング等による並列性の拡大により、スカラー型、ベクトル型計算機で並列性の利用度が向上する。さらに、演算/ロード・ストア比を向上し、データの並び替えでメモリ参照の時間的、空間的な局所性を向上させ、さらに無駄なデータのストアやロードを削除すれば、実効的なデータの供給性が向上し、プロセッサの性能が向上する。FFBでの性能向上は、このような並

列性の統合的利用、データの局所性の統合的な最適化によりもたらされたものである。

PHASE: ソフトウェアにおいて徹底的に非並列部を削減することは、並列性向上において重要である。問題の規模や利用するプロセッサ数の増大により、ソフトウェア内の非並列部の負荷が徐々に顕在化する。この現象は、ある程度並列最適化されたコードで小中規模の計算機環境では目立たず、むしろ、大規模計算資源で、高並列での問題を取り扱うときにその重要性が明らかになることがある。

PHASEの場合、原子数とプロセッサ数を増加させて性能データを分析することにより、原子についてのループ内で、原子に関する非並列計算があることが分かった。原子数が少なく少数プロセッサの時は、その計算量は性能に対して問題にならないが、原子数、プロセッサ数を増やすと性能に対する影響が無視できなくなる。第一原理計算では、原子近傍のポテンシャルを偽ポテンシャルで表し、計算量を激減させる方法を取ることが多

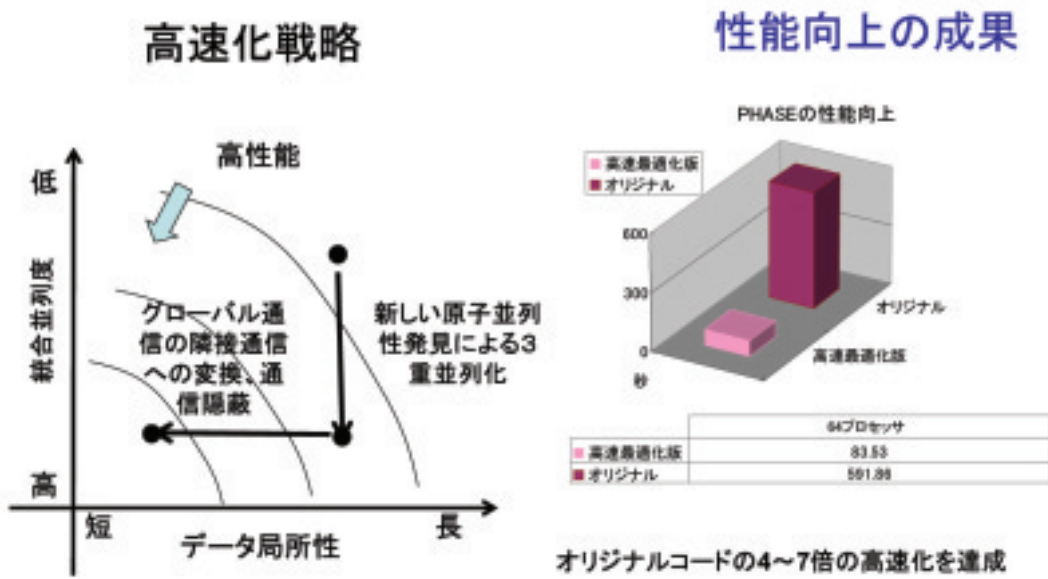


図-8 PHASEに対する統合的高速化技術の応用例

く、PAHSEでも同様である。PHASEではエネルギーレベルに関しても並列化している。各エネルギーレベルの計算には全原子の偽ポテンシャルに関する情報が必要なため、各エネルギーレベルの計算を担当する各プロセッサは全原子に関するループを持つことになる。ところが、この全原子に関して計算される情報が、各エネルギーレベルで共通に必要なため、エネルギーレベルで並列化しただけでは、この原子に関する計算が冗長に計算され非並列計算となる。つまり、各プロセッサで全原子に関する同じ計算をするという無駄を生じていたのである。

我々は、この原子に関する計算が並列に計算できるという原子並列性を発見した。しかし、原子を各プロセッサに振り分けて計算させると、その計算結果を全プロセッサが全プロセッサへ通信し、また受信するという（グローバル通信）非常に通信負荷が高い処理が必要となる。

そこで、各プロセッサに割り当てられてられた原子に関する計算情報を各プロセッサ間でいわゆる「バケツリレー」的に隣へ渡す（隣接通信）方式を取り、通信負荷を低減した。

これは、データのアクセス距離の短縮化でもあり、ある意味では局所性の向上とも考えられる。PHASEの性能向上は、新しい並列性の発見、かつ通信方式の変換による実効的なデータ局所化の例である（図-8）。

(2) 性能推定の結果例

計算カーネル群へのコードの分解、また各計算カーネルの大規模・高並列での性能データの取得、さらに処理時間、並列性能、単体演算性能、通信負荷などを分析し、その結果をもとに、計算量、通信性、性能等を推定する数理モデルを構築した。以下、構築例を紹介する。

FFB：本コードは乱流の非定常で複雑な動きを正確に予測する方法として、Large eddy simulation (LES) の実用化を目指している。FFBは有限要素法を用いた流体計算用プログラムである。有限要素法には全体剛性マトリクスを構築する計算法と全体構成マトリクスを構成せずに要素剛性マトリクスのみで計算を進めるエレメント・バイ・エレメント法がある。FFBは後者の方法を採用し

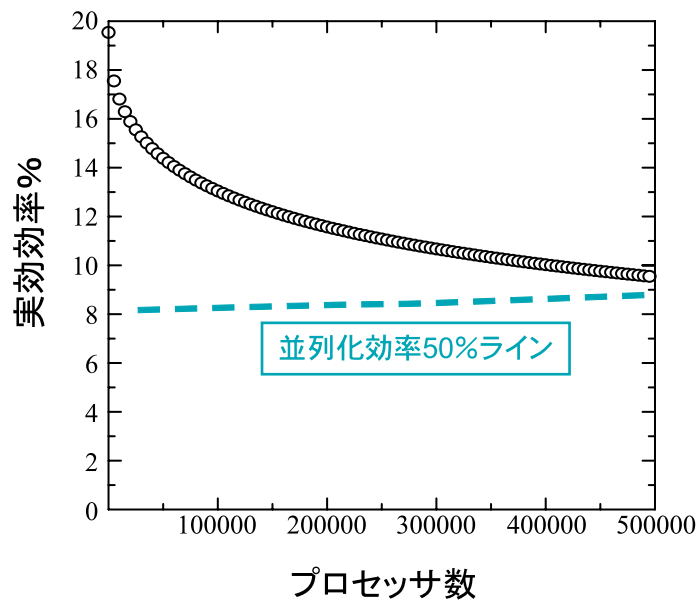


図-9 FFBに対するペタスケールシミュレーションでの性能推定

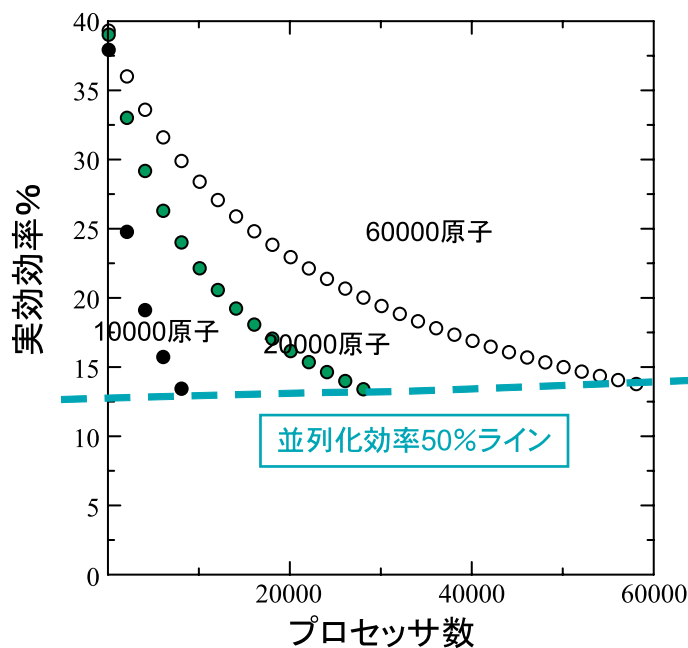


図-10 PHASEに対するペタスケールシミュレーションでの性能推定

ている。FFBでは、速度場の計算、圧力場の計算が主である。速度場の計算ソルバーとして、BCGSTABとヤコビ法が使用され、圧力場では、BCGSTABとABMAC法が使用される。FFBでは、要素毎にマトリクス・ベクトル積を計算した後、要素毎の節点データをグローバルな節点データに足し込む処理が主要

な計算である。

FFBのブロック化された計算カーネル群の中から (A) 圧力に関する計算、(B) フラックスの計算、(C) 移流拡散項の計算、(D) 節点ベクトルの計算、(E) 内積計算の5つのカーネルを主要計算カーネルとして特定した。性能データ測定は、1プロセッサ当た

り、30万要素程度の規模で、弱スケーリング（各プロセッサの要素数を一定に固定）にて、2 / 4 / 8 / 16 / 32 / 64 / 128 / 256 / 512 / 1024 / 2048のプロセッサ数で、地球シミュレータを使って性能データを計測した。この性能データと物理数学モデル、解析モデルを組み合わせ、3(2)のモデル式を元に性能推定モデルを構築した。この性能推定モデルを使い500,000プロセッサまでの実効効率を推定した例を図9に示す。FFBは工学コードであるため、問題の複雑化、精度向上での性能が問題となる。そこで弱スケーリングでの性能推定を実施し、並列化効率50%のラインまでの実効効率を計算した。

PHASE：本ソフトウェアは、方程式を解くために物理量を関数展開する基底として、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比較的精度良く電子状態計算を求めることができる。PHASEの計算処理は、まず初期値設定後、波動関数をKhon-Sham方程式を固有値問題として解き更新し、その後電荷密度を更新し、電荷密度分布を反復計算にて求める。その後、原子に働く力を計算し、原子位置を更新し、原子に働く力がゼロとなりエネルギーが最低となる安定結晶構造が得られるまで反復計算を行う。

PHASEの主要計算カーネルとして、ブロック化された計算カーネル群の中から、(A) 波動関数の直交化計算、(B) 原子に働く力の計算、(C) 非局所項の計算、(D) 残差ベクトルの計算、(E) 修正ベクトル計算の5つのカーネルを特定した。性能データ測定は、原子数512~8,000、プロセッサ数128~4,096と地球シミュレータの全計算資源を使った大規模、高並列シミュレーションで実施した。この大規模性能測定データと物理数学モデル、解析モデルを組み合わせ、3(2)で示した式を元に性能推定モデルを構築し

た。この性能推定モデルを使い60,000プロセッサまでの推定実効効率を求めた例を図10に示す。原子数をパラメータとし、並列化効率50%のラインまでの実効効率を推定したものである。

今回、構築した性能推定技術により、FFBとPHASEについて、ペタスケールレベルの問題規模、プロセッサ数での性能を試験的に推定した。今後、本結果をより大規模な問題とプロセッサ数での実測データにて検討、検証し、その結果を元にモデルの精度向上、本性能推定技術の高度化を進める予定である。また、他の応用ソフトウェアについても適用を拡大する予定である。

5. まとめ

ペタスケールの大規模シミュレーションを見据えた科学技術応用ソフトウェアの高速最適化技術の研究について概要を紹介した。本高速最適化技術をFFBとPHASEに適用しその効果を確認した。今後、適用する応用ソフトウェアの範囲を拡大するとともに、使用するスーパーコンピュータの範囲も拡大し、高速最適化技術の高度化を進める予定である。大規模計算科学は、従来の計算機能力では扱えない研究開発課題について、高速演算性が解の到達時間を早め、パラメータの探索空間、さらにシミュレーションの時間空間を拡大することにより、新発見や新技術開発を可能にする有効な手段として認識されている。よって、高速最適化技術の重要性は今後より一層増していく。かつ、高速最適化技術は、物理・数学モデルからスーパーコンピュータまでを総合的に理解、駆使するものであり、幅広い知識と経験さらに技術を必要とする分野である。しかも、IT技術の進展を俯瞰すれば、組み込みソフトウェアなど多方面に高度な応用性を潜在的に有している分野でもある。本稿が多くの若手のみならず熟練の研究

者、技術者の目に留まり、優秀な人材がこの分野に集い活躍されれば幸いである。

謝 辞

本研究は、文部科学省ITプログラム「戦略的基盤ソフトウェアの開発」「戦略的革新シミュレーションソフトウェアの研究開発」の成果に基づくものである。本研究では、プロジェクトオフィサーの東京大学生産技術研究所小林敏雄前教授〔現、(財)自動車研究所

長〕、プロジェクトリーダー加藤千幸教授、地球シミュレータセンターの平野哲部長、東京大学情報基盤センターの金田康正教授など多方面からのご支援を頂いたことに深く感謝の意を表す。

参考文献

[1] 文部科学省ITプログラム「戦略的革新シミュレーションソフトウェアの研究開発」成果報告会発表資料集、2008年3月。