TSUBAME 共同利用　平成 27 年度　学術利用　成果報告書

利用課題名　高性能・高生産性を達成する垂直統合型アプリケーションフレームワーク
英文：High Performance, Highly Productive Application Frameworks

丸山直也
Naoya Maruyama

理化学研究所計算科学研究機構
RIKEN Advanced Institute for Computational Science
http://mt.aics.riken.jp/

本課題ではポストペタスケールに向けた最重要課題である「並列性の克服」、「信頼性」、「低消費電力化」の解決に大きく貢献する高生産性垂直統合型ソフトウェアスタックの研究開発を行う。これはスケーラブルマルチスレッドランタイムを基盤としたドメイン特化型アプリケーションフレームワークであり、自動並列化、自動チューニング、耐故障性、電力最適化等の各種技術を透過的に内包する。我々は提案ソフトウェアスタック構成法を流体シミュレーション（CFD）および分子動力学法（MD）を対象として設計し、それらを最新の大規模ヘテロジニアススーパーコンピュータである TSUBAME2 を基盤として設計開発する。

We develop a vertically-integrated highly productive software stack for achieving parallel, resilient, and power-aware application development. We design the software stack as a domain-specific application framework built on top of a scalable multithreading runtime. More specifically, we develop frameworks for computational fluid dynamics and molecular dynamics applications using large-scale heterogeneous supercomputers such as TSUBAME2.

Keywords: 5つ程度 GPGPU, Programming model, high-level framework

背景と目的

GPU などのベクトルアクセラレータを共用したヘテロジニアスシステムの重要性は消費電力最適化の重要性と共に今日のペタスケールシステムにおいて広く認識されており、今日の最新 GPU は数千の SIMD コアが搭載されたメニーコアプロセッサであるが、このような多数コアの超高密度実装はポストペタスケール、さらにはエクサスケールの実現に向けてさらなる性能向上に不可欠な技術である。実際に主要プロセッベンダは小規模スカラーコアと大規模ベクトルアクセラレータを統合したヘテロジニアスプロセッサの設計開発を行っていると言われており、2015 年頃のポストペタスケール、2018 年頃のエクサスケールにおいて主流となることが予測される。

上述のアーキテクチャのためのソフトウェアスタックはヘテロジニアスプロセッサのプログラミング、性能最適化、スケーラビリティ、耐故障性、電力効率最適化などの種々の課題を解決し、またそのプログラミングモデルが将来に渡って連続性のある高い生産性を有したものでなければならない。今日におけるヘテロジニアスシステム向けソフトウェアスタックではシステムの性能を最大限に引き出すためには複雑かつ煩雑なプログラミングが必要とされ、また将来のアーキテクチャの革新に対する連続性を有しない。さらにソフトウェアスタック全体としては上述のプログラミングモデルの問題に加えて、通信オーバーヘッドの隠蔽や動的負荷分散によるスケーラビリティの達成、耐故障性、電力効率の最適化を実現しなければならないが、そのようなソフトウェアスタックは我々の知る限り存在しない。

本プロジェクトでは、上述したポストペタスケールシステムにおける課題を同時に解決する、新たな垂直統合型ソフトウェアスタックを提案する。ターゲットとするアプリケーションドメインを限定し、そのドメインに対して最適化したアプリケーションフレームワークをシステムソフトウェアおよびアプリケーション研究の専門家の密な協力により設計開発することで、上記課題を高い生産性のもと解決する。

概要

ポストペタスケール時代に向けた高性能・高生産性 AMR フレームワークとして、GPU クラスタ向け Octree による AMR フレームワークに取り組んだ。これは構造格子差分法を対象とし、AMR 機能をランタイムライブラリとして実現している。本稿執筆時点ではランタイム部の実装を終えており、Octree のリーフに相当するセルに対するステンシル計算やセルを refine するか coarsening するかの判断はユーザから与えられた CUDA 関数を呼び出す実行方式としている。AMR ランタイムでは典型的に用いられる補完方式や格子間隔の調整方式を実装しており、プログラマはそれらを取捨選択してプログラミング可能である。ランタイムはユーザから与えられた関数を基に適切に格子の refinement および coarsening を実施しつつ計算を進め、その際に必要となるホスト・GPU 間のデータ転送や GPU メモリの管理などはランタイム側によって自動的に管理される。Gamer や Uintah 等の既存の GPU をサポートした AMR ではステンシルの計算のみ GPU を用い、AMR 処理は CPU 側で実装する方式が主流だが、我々のランタイムは CPU・GPU 間のデータ転送を最小化するためにすべての処理を GPU 側で実現している。また、複数ノード実行にも対応しており、空間充填曲線による Octree の分散メモリにまたがる分割および隣接データの MPI による交換もランタイム側で自動的に処理される。

上記と同様に差分法等の計算を対象とした最適化としてカーネル融合の自動化による最適化手法を開発した。これは構造格子ステンシル計算の重要な応用例である気象・気候モデル等の特徴を応用したものである。そのようなアプリケーションでは単一のステンシルを時間方向に繰り返し実行するのではなく、単一時間ステップに数十もの多数のステンシルを計算するパターンが一般的である。このような場合にはステンシルを適切に融合もしくは分割することが局所性最適化に有効だが、一般的なコンパイラや既存最適化手法では最適化対象問題空間が指数的に拡大するため実現されていない。 我々は探索ヒューリスティックスを応用することでこれまでの課題を解決し、数十、数百におよびステンシルカーネルの融合・分割を現実的な時間にて求める自動最適化手法を設計、開発した。具体的には、まず CUDA プログラムとして与えられるステンシルカーネルのプログラム解析によりその性能モデルを構築し、またプログラム全体のカーネルの依存グラフを計算する。次に融合することによってオンチップメモリを介したデータの再利用が可能になるカーネルの組み合わせを全カーネルについて計算し、それぞれ性能モデルにより融合の効果を推定し、最良組み合わせを探索する。カーネルの組み合わせパターン数はカーネル総数に対して指数関数的に増加するため、総当たりによる探索は非現実的であり、我々は進化計算を応用したヒューリスティックスにより数百のカーネルに対しても数分で探索を終了可能なアルゴリズムを開発した。

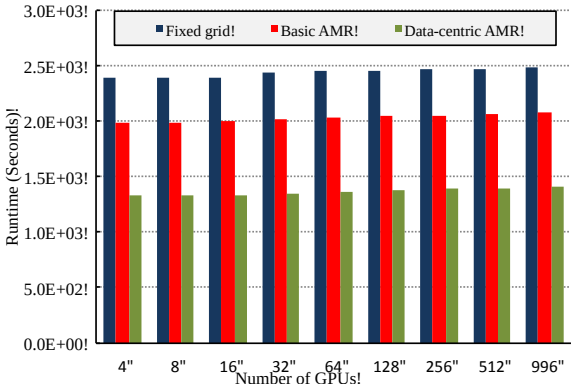

図 1 Phase-Field 法の AMR による性能比較結果

結果および考察

上述の AMR ランタイムの評価として GAMER との性能比較を hydrodynamics、shallow-waters シミュレーションにて行い、CPU・GPU 間データ移動を削減した効果により最大 1.4 倍の性能向上を確認した[(1)-84]。また、東工大青木研究室等にて開発されたゴードンベルを受賞した Phase-field 法コードの AMR 化を実施し1000GPU 弱までのウィークスケーリング評価を行い、必要な精度を保ちつつ AMR により性能を 1.6 倍に向上可能であることを確認した（図 4）。また AMR 化に必要なプログラム行数はたかだか700行程度でありランタイムにより AMR の実現を大幅に簡易化できた。

カーネル融合最適化の評価として提案手法を 6 本の実際の GPU アプリケーションに適用した。それぞれほぼ数分で最適解に到達可能であり、実際に 1.1 倍から 1.7 倍の性能向上を達成した。

まとめ、今後の課題

（まとめと今後の課題について記載してください。）。

本課題では高性能・高生産性を達成するソフトウェアスタックの実現に向けて、AMR フレームワークの設計、開発を実施した。また、ステンシル等の計算を対象とした高度な最適化として大規模プログラムに適用可能なカーネル融合・分割アルゴリズムを開発し、その評価を行った。これらのソフトウェアについては次年度以降も継続して開発を続けていく予定である。