TSUBAME 共同利用　平成 25 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/

邦文抄録（300 字程度）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。

英文抄録（100 words 程度）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

上述の目的を達成するために本課題では流体シミュ

レーション向けフレームワークである Physis の設計、開発を中心に進めた。Physis は C 言語を拡張したステンシル計算を簡便に記述可能とするドメイン特化型言語およびそのランタイムから構成される。Physis にて記述したステンシル計算は Physis トランスレータによって各種実行アーキテクチャ向けコードに変換され、プログラマが明示的に並列化や GPU プログラミングを行う必要はない。本課題では Physis の設計の拡張やトランスレータの最適化等を進めた。また Physis の実アプリケーションへの適用評価を進めた。

また、より高度な最適化手法の開発、評価を進めた。GPU 上で実行されるプログラムであるカーネルは複数のカーネルを融合することによりデータ再利用性を高めることが可能である。これは CPU 向けプログラムでは性能最適化のひとつであるループ融合として同様に開発されてきた技術である。我々は GPU カーネルの最適な融合を性能モデルにより決定する手法を提案しており、本課題において TSUBAME を用いた評価を進めた。

また、同様のフレームワークの例として CPU 用コードと GPU 用コードをディレクティブにより同一コードに共通的に埋め込み適宜選択可能な Hybrid Fortran フレームワークの開発および評価も進めた。これは気象庁との共同研究であり、次世代気象モデルとして開発が進められている ASUCA の物理過程を対象として進めた。

結果および考察

本課題では Physis の設計、実装として GPU 向けコード生成の最適化をすすめた。現在では Fermi 世代の GPU に対しては最適に近い性能が達成できている。Kepler 世代の GPU については最適化手法の基礎的な開発、評価を行い、フレームワークによるコード生成のための知見を得た。

カーネル融合最適化については気象シミュレーションアプリケーションを題材として開発を進めた。本アルゴリズムでは数十から数百と多数のカーネルから構成されるアプリケーションを対象とする。一般的にはそのような多数のカーネルの集まりについて最適な融合を決定することは困難である。これは融合により GPU リソースの枯渇が生じ、かえって性能が低下すること、従って融合するカーネルの組み合わせをカーネルの利用リソースやコードの特徴を考慮したうえで決定する必要があるためであり、カーネル数が増加した場合は特に組み合わせ問題として最適組み合わせを求めることが困難になる。従って現実的には人手による最適化の場合は単純な組み合わせの融合のみを考慮する程度に限られることが多く、従ってそれによる性能向上もまた限定されがちである。我々はこの問題に対してカーネル融合を組み合わせ最適化問題に帰着させ、数百のカーネルの集まりに対しても現実的な時間で最適解を求めることができる手法を開発した。本手法は融合による効果を推定するために性能モデルを構築する。性能モデルはカーネルの命令アクセス回数などから構成される。我々は本アルゴリズムを気象アプリケーションの CUDA 実装に適用した。その結果、カーネル融合により 15%の性能向上を達成し、かつその組み合わせの探索を数時間で完了可能であることを確認した。

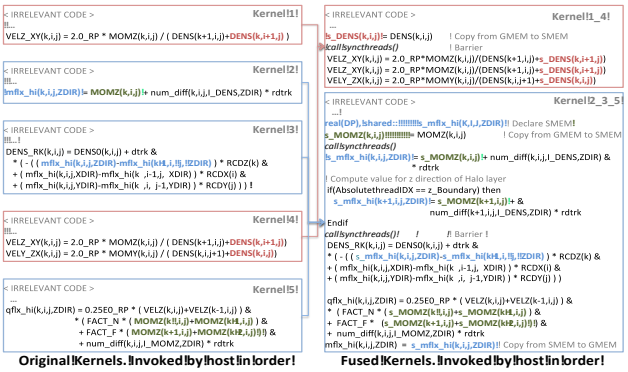

図 1 カーネル融合の例

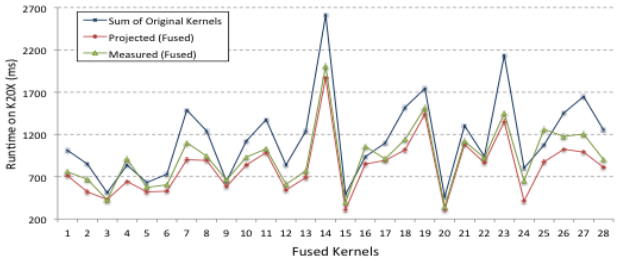

図 2 カーネル融合による実測と性能モデルの比較

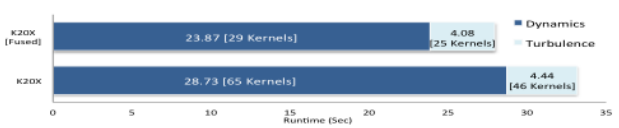

図 3 カーネル融合による性能向上

Fortran 向けフレームワークである Hybrid Fortran

については ASUCA の物理過程の移植を行い、これにより既存の ASUCA 力学過程 GPU 実装とあわせて次世代気象モデル全体が GPU 実行可能となった。

まとめ、今後の課題

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、その一要素技術であるステンシルフレームワーク Physis の設計、実装を進めた。また、さらなる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アルゴリズムを開発し、その評価を行った。Fortran 向けフレームワークでは ASUCA の物理過程の移植を完了し、その適用可能性を示すことができた。

今後は開発したカーネル融合アルゴリズムの改良、特に性能モデルの改良や探索アルゴリズムの改良を検討している。また、新しい世代の GPU への対応や他のアクセラレータアーキテクチャへの対応を検討中である。