

TSUBAME 共同利用 平成 26 年度 学術利用 成果報告書

利用課題名 GPU を用いた気象・気候アプリケーションライブラリの開発  
英文: Development of library for weather and climate model using GPUs

利用課題責任者 八代尚  
First name Surname Hisashi Yashiro

(独)理化学研究所 計算科学研究機構  
RIKEN Advanced Institute for Computational Science  
<http://www.aics.riken.jp/>

#### 邦文抄録

本課題は、大規模かつ複雑化した気象・気候アプリケーション(気象モデル)の演算加速機構(アクセラレータ)最適化を進め、大規模並列計算を達成することを目的としている。気象モデルの主要コンポーネントの一つである大気放射過程のコードを題材として、OpenACCを用いた GPU への最適化を行った結果、メモリバンド幅に見合う演算性能を得た。大気放射は3次元格子のうち大気鉛直方向に依存性を持ち、配列次元の順序が演算性能に無視できない差をもたらすことが確認された。配列次元の頻繁な入れ替えは無視できないコストになるため、アプリケーション全体でデータの並び順を包括的に管理する必要がある。

#### 英文抄録(100 words 程度)

We applied GPU optimization to physics library of weather/climate application by using OpenACC. The kernel code of atmospheric radiation module showed reasonable speedup on the GPGPU compared to the case of CPU-only simulation. The radiation code has dependency in the vertical direction of the three dimensional grid and array order of horizontal/vertical grid is critical parameter for GPU calculation. The optimization and the evaluation between two types of data layout showed tradeoff between the code portability and performance.

*Keywords:* Weather and climate model, Memory-bound application, OpenACC, GPGPU

#### 背景と目的

気象・気候モデルは計算機を用いた現象の解明のために、まだまだ多くの計算機リソースを必要とする分野の一つである。一般的に気象モデルはメモリバンド幅律速のアプリケーションであり、プロダクト実験においては演算器からネットワーク、ファイル I/O までのすべてのステージで高いデータ転送性能を要求する。モデルのソースコード規模は数十万行と大きく、ある1区間だけで演算時間と演算量の大部分を占めるような計算カーネルが存在しない「flat profile」という特徴を持っている。今後気象モデルが電力消費を抑えたまま、より大規模で高速なモデルシミュレーションを行うために、GPGPUを用いたヘテロジニアスな構成のスーパーコンピュータを積極的に利用していく必要がある。近年、ディレクティブベースでのアクセラレータ利用を可能にするプログラミング規格として、OpenACC が登場した。これにより、CUDA C/Fortran などを用いたソースコード

の大幅な書き換えを行うことなく、既存のソースコードにディレクティブを追加することによって CPU-アクセラレータ間のデータ転送とアクセラレータ上での演算実行が実現する。

本課題は、既に計算科学の第一線で活用され、大規模化・複雑化した気象モデルに対し OpenACC を適用することによって、コードのポータビリティや可読性を失うことなく、アクセラレータを利用しその性能を引き出すことが可能であるかを評価することを目的とする。

#### 概要

大気放射過程は大気モデルの主要なコンポーネントの一つである。太陽から入射した短波放射は大気中のガスによって吸収され、雲やエアロゾルによって反射・散乱される。残りは海面や陸面によって反射・吸収され、地表面を温める。温められた地表面や大気は赤外線を射出し、この長波放射もまた吸収・反射・散乱を経て大気圏外へ射出される。大気放射過程はこのような大気

中の放射伝達に関わる過程をモデル化したものである。多くの気象モデルでは放射に対して平行平板大気を仮定しており、3次元の離散化格子に対して水平方向には独立し、鉛直方向に依存性を持つような計算が行われる。放射伝達計算は紫外域から可視、赤外域までの波長全域について、非常に狭い波長幅ごとに計算を行う必要があるが、気象モデル内で用いるには計算量が多すぎるために、相関  $k$  分布法などを用いることで、波長方向の演算量を削減したブロードバンド放射モデルが採用されている。本課題では東京大学で開発されたブロードバンド放射伝達モデル `mstrn`(Sekiguchi and Nakajima, 2008)に対して、OpenACC ディレクティブを適用し、GPU で高い性能を出すために必要なコードの変更とそのポータビリティについて調査を行った。

利用したソースコードは理化学研究所が主導して開発を進めている気候気象のための基盤ライブラリ SCALE (Scalable Computing for Advanced Library and Environment, Nishizawa et al., 2015, <http://scale.aics.riken.jp/>より入手可能)に収録されているものを用いた。これはオリジナル版の `mstrnX` (<http://ccsr.aori.u-tokyo.ac.jp/~clastr/>より入手可能)を元に、演算最適化や気象モデルに合わせた配列次元の拡張などを施したものである。SCALE ライブラリは、ライブラリを利用するアプリケーション(気象モデル)が大気の3次元格子を持っていることを想定して、 $(k,ij)$ の順の配列次元を持っている。このうち、 $k$  が鉛直方向の格子、 $ij$  が水平方向の格子を表す。

結果および考察

〈OpenACC の適用とソースコードの変更点〉

OpenACC ディレクティブの適用と最適化は NVIDIA 社の技術者の全面的なサポートを受けて行われた。基本的な最適化の指針を以下に示す。これらの手法は H25 年度に行われた全球非静力学 NICAM の力学コア最適化の成果を反映している。

- ・ メモリアロケーションと演算部分の分離の徹底: メモリ確保の必要な配列はセットアップでアロケーションを行い、演算部分から排除する。時間発展しない配列はセットアップ時に `present_or_copyin` 節を用いて転送を行う。
- ・ カーネルの非同期実行: 各ループに対しカーネル

節を適用する際に、出来る限り `async` 節を付加した実行を行う。

- ・ 放射フラックスを大気上端から地表面まで順に計算し、再び逆方向に計算を行う部分に関しては、鉛直方向にループの依存関係が存在する。GPU 上での計算のために $(k,ij)$ の配列を $(i,k,j)$ の次元順を持った配列に一時的に格納し、演算を行う。

配列次元入れ替えに関しては、スカラー機での最適化方法と大きく異なるため、ソースコードの共有は出来ない。これはソースコードのポータビリティを損なうものである。そこで本課題では、はじめから配列次元がすべて $(ij,k)$ であるようなバージョンも作成し、同様に性能評価を行った。

〈演算性能〉

上の変更点で述べたとおり、最内の配列次元のデータに依存性がある場合には、もとのソースコードに OpenACC ディレクティブを挿入しただけでは期待される性能を得られなかった。メモリアクセスの最適化のために鉛直依存のある変数の計算とない計算を分離し、次元順の異なる配列の一時利用を行ったところ、OpenACC 適用直後と比べておよそ5倍の性能を得た。特に鉛直依存があり、演算量の多いループの場合は、時間コストをかけてでも `read` される配列の詰替えを事前に行なった方が高速化する結果となった。図1に、CPUのみでの実行とGPUを用いた実行での計算速度の結果比較を示す。結果より、GPUを用いた実行は

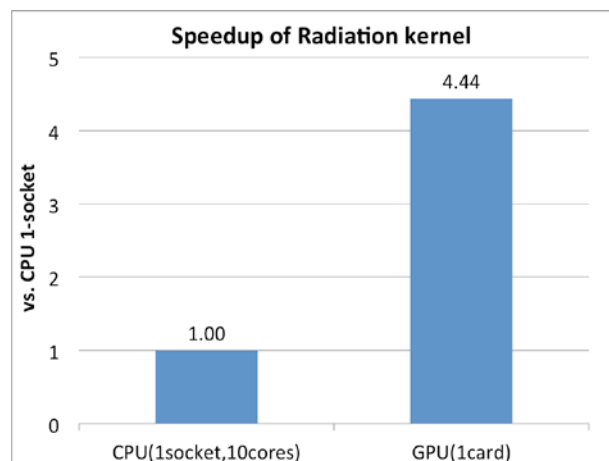


図1: 大気放射コードの演算カーネルの実行速度比較結果。1ソケット CPU(Intel Xeon E5-2690 v2,3.0GHz,10 core)での演算速度を1としたとき(左)の、GPU 1枚(NVIDIA Tesla K40)を用いた演算速度の倍率(右)を示す。値の大きいほうがより高速である。

CPU よりもおよそ 4.4 倍高速である結果を得た。CPU、GPU 実行においてどちらもメモリ転送性能を十分に利用できており、ディレクティブによるプログラミングスタイルをとる OpenACC が十分に性能の高い実行コードを生成出来ていることが確認されている。今回用いたブロードバンド放射伝達モデルのコードは、大気の流れ力学計算を行うステンシル計算部分よりも演算密度が高く要求する B/F 比は低い。そのため演算律速となりメモリバンド幅の影響は受けにくいと一般的に認識されている。しかし実際は、行列積のようなキャッシュ内での値の使い回しが効く演算ではなく、演算量の多い指数・対数計算を(3次元格子) $\times$ (波長帯)に対して順に適用する部分が多くデータの再利用性が低いため、最終的にメモリバンド幅に律速される形になる。本結果においても、実行速度の GPU/CPU 比は演算性能の比(6.0倍)よりも、メモリ転送性能の比(4.8倍)によるものが大きいと考察される。

図2に、サブルーチン全体のデータ構造をそれぞれ(k,i,j)、(i,j,k)の順にした場合に、OpenACC ディレクティブの適用と最適化を施したカーネルコードの実行時間比較結果を示す。(i,j,k)版のコードでの最適化方針は(k,i,j)版とほとんど変わらず、メモリアクセス最適化のための追加的な配列詰替えが不要な分だけ高速化した形となった。放射伝達モデルで用いられる配列の多くは気象・気候モデル全体で共有して用いられる変数(気温や気圧、水蒸気量等)であり、配列の次元順はモデル全体

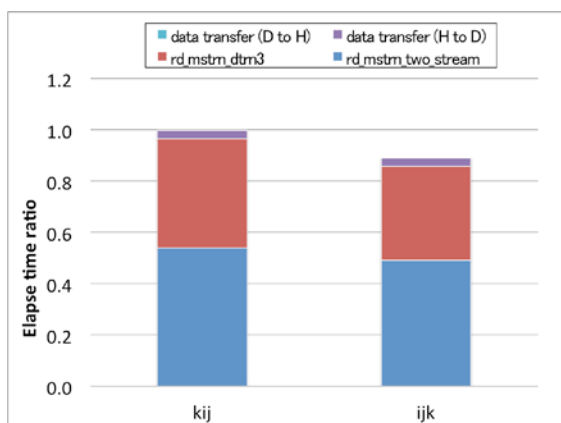


図2: 大気放射コードの演算カーネルの GPU を用いた実行時間比較結果。サブルーチン全体のデータ構造について、鉛直方向の次元(k)が前(左)の時の演算時間を1として、水平方向の次元(i,j)が前(右)の時間を示す。値の小さい方がより高速である。どちらも OpenACC を用いた GPU 最適化を行ない、それぞれに適したチューニングを施した。

のデザインに関わる要素である。配列次元の頻繁な入れ替えは演算密度の低い気象・気候モデルにとって無視できないコストになるため、アプリケーション全体でデータの並び順を包括的に管理する必要がある。GPU やベクトル機を用いた計算を念頭においた場合、データレイアウトは水平格子が前に来ることが望ましい。しかしスカラー機と OpenMP の組み合わせでの性能向上を目指した場合、鉛直格子が前に来るレイアウトが優先される。水平格子を分割して袖領域をノード間通信する場合も、鉛直格子が前に来ている場合は通信のために袖領域の格子を連続メモリ領域にコピー(パッキング)する必要がなく、直接指定して送ることが可能になる。このようなデータ構造の選択はトレードオフの関係にあり、計算機に合わせた最適化が必要であるといえる。その上で、コードのポータビリティを保つためには、データ構造の切り替えを容易にするソースコード変換や抽象化言語での記述が必要となるであろう。

最適化されたカーネルを含むアプリケーション全体での評価については、PGI コンパイラを用いて整備した HDF5, NetCDF 等のファイル I/O ライブラリ周りで異常終了するなど解決の難しいトラブルが続いたため、本課題期間内に TSUBAME2.5 上での並列 GPU 計算をするまでには至らなかった。今後は OpenACC の適用をさらに物理諸過程の各コンポーネントに拡張し、気象気候モデルのフルパッケージでの GPU 計算に向けた取り組みを進める予定である。

## 参考文献

- [1] Sekiguchi, M., and Nakajima, T. (2008) A k-distribution-based radiation code and its computational optimization for an atmospheric general circulation model.; *J. Quant. Spectrosc. Radiat. Transfer*, **109**, 2779–2793.
- [2] Nishizawa, S., Yashiro, H., Sato, Y., Miyamoto, Y., and Tomita, H. (2015) Influence of grid aspect ratio on planetary boundary layer turbulence in large-eddy simulations, *Geosci. Model Dev. Discuss.*, **8**, 6021–6094, doi:10.5194/gmdd-8-6021-2015.