TSUBAME 共同利用　平成 25 年度　学術利用　成果報告書

利用課題名　GPU を用いた全球高解像度気象モデル力学コアの開発
英文：Development of GPU-based dynamical core for high-resolution GCM simulation

富田浩文
Hirofumi Tomita

（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。全球高解像度モデル NICAM に OpenACC を適用することにより、最小限のコード追加・変更で力学コア全体を GPU 上で計算することが可能になった。演算性能はメモリバンド幅に見合うものであり、高レベル言語である OpenACC が演算性能と生産性の両面で有用であることが実証された。

英文抄録（100 words 程度）
Simulation using a large number of accelerators is a big problem for weather / climate application. The scale of application codes is large and complicated; it costs in rewriting using a low-level language. We applied GPU optimization to global high-resolution atmospheric model, NICAM, by using OpenACC. The whole part of dynamical core of NICAM was calculated on the hundreds of GPGPUs with reasonable performance efficiency. The result demonstrates that OpenACC is promising tool in both performance and productivity.

Keywords: Global Circulation Model, Memory-bound application, OpenACC, GPGPU

背景と目的

台風の発生過程解明や気候システムの諸要素の研究に代表される学術的な側面だけでなく、天気予報や気候変動予測のような人間社会と密接に関わる要件においても、気象・気候の数値シミュレーションは大きな役割を果たしている。大気の小さなスケール(数m)から全球スケールの現象までを統合的に扱い、スケール間の相互作用をよりよく表現するには、現在よりもさらに細かい空間解像度でのシミュレーションを実現することが求められている。これはさらに強力なスーパーコンピューターを活用する必要があるということである。一方、近年のスーパーコンピューターの演算性能は GPGPU やメニーコアコプロセッサのようなアクセラレータに依る部分が多く、その比率は増加の一途を辿っている。このような背景から、気象モデルも積極的にアクセラレータを利用することが求められている。気象モデルは要求する Byte/FLOP 比が一般的に高く、アーキテクチャの演算性能よりもメモリバンド幅に律速される。この点でも、CPU より高いメモリ転送性能をもつアクセラレータの利用は有効であると考えられる。しかし、気象モデルのソースコード規模は非常に大きく（例えば、NICAM は数 10 万行）、またモデル自体が大気力学、雲微物理学、接地微気象学、陸水学、大気放射学などの様々な分野で研究開発されている小モデルの集合体である。そのためソースコードのメンテナンスや大規模な書き換えには大きな困難を伴う。

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無く、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

実アプリケーションとして全球雲解像モデル NICAM[1][2]の流体力学ソルバー部分（力学コア）を中心にパッケージングされた NICAM-DC を用いて、OpenACC を用いた GPU 最適化の実施と演算性能、スケーリング性能の評価を行った。NICAM は正20面体格子配置と有限体積法を採用した、主に全球大気シミュレーションのためのモデルである。NICAM はその高並列性と演算性能によって高解像度化を達成し、雲システムや積乱雲ひとつひとつを直接表現する「雲解像モデル」としてこれまでに多くの科学的成果を上げている。プロセス並列には MPI を用い、OpenMP を用いたスレッド並列は行っていない。

本課題では、主要演算部分に 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。加えて、TSUBAME2.5 の CPU、GPU を用いた際の演算性能やスケーラビリティ、電力性能についての評価を行った。NICAM-DC は CPU、GPU での実行において、それぞれメモリバンド幅に見合う演算性能を発揮し、GPU の高いメモリ転送性能と電力性能を十分に活用出来ることが示された。スケーラビリティについても良好なウィークスケーリング性能を示した。ストロングスケーリングについては、問題サイズの減少に伴い、GPU を利用した際に急速に高速化率が頭打ちになることがわかった。これはネットワーク通信にかかる時間に加え、GPU-Host 間のデータ転送にかかる時間の割合が、演算時間に比べて増大することが大きな要因となっている。

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は以下の通りである。

- メモリアロケーションと演算部分の分離の徹底：GPU 上で計算するために必要なメトリクス項などの配列は、必ずセットアップ部分でメモリ確保を行い、演算部分から排除する。
- 更新されない配列はすべて GPU 上に常駐させる：上記の配列は時間発展しないので、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、袖通信のタイミングやデータ依存性のある部分を明確にし、極力 async 節を付加した実行を行う。
- 袖通信の最適化：NICAM の格子配置は正20面体を用いており、隣接通信の相手や格子位置は複雑である。そのため、袖通信の際に送るデータは必ず pack/unpack を行っている. この作業を GPU 上で行い、ネットワーク通信に必要なデータのみを Host 側に転送し，再受け取りを行う。
- 特異点格子の計算：マスターノードで行われる特異点格子の計算は演算量が少ないため GPU には転送せず、CPU で行う。CUDA 等を用いた場合、通常の格子と特異点の計算カーネルをサブルーチン単位で分離することになり、メンテナンスが困難になることが予想されるが、OpenACC の場合は同じソースコードに共存することが出来る。
- **ヒストリ出力の扱い：気象モデルでは解析に用いる2次元・3次元変数の時系列を取得するため、途中の値をファイルに出力することが一般的である。このヒストリ出力のための鉛直内挿、時間方向の平均化や診断値の計算はすべて GPU 上で行い、数十～数百ステップごとのファイル出力のタイミングで Host 側に転送し、書き出しを行った。**

NICAM のデータ構造は(ij,k,l)=(水平格子,鉛直格子,領域分割単位)であり、多くの場合で水平格子が最もサイズが大きい。GPU 上での実行に際し、このデータ構造を変更する必要はなかった。ただし、いくつかのステンシル演算カーネルではスカラー機用のキャッシュラインチューニングとして、小さいサイズの配列次元を最内に変更した箇所があったため、OpenACC 適用時に再度変更された。これは、アーキテクチャによってデータ順序の異なる AoS 版と SoA 版の2バージョンを使い分ける必要があることを示唆している。しかし、切り替えるコードの分量としてはそれほど大きくないため、メンテナンス性の観点から見ても十分に対応可能であると判断される。以上のことをふまえ、挿入されたディレクティブ行および元コードに変更が施された行は 2000 行程度であった。これは総コード行数 58000 行の5%に満たない量であり、十分なポータビリティとメンテナンス性を保持したまま、GPU への対応が完了したといえる。

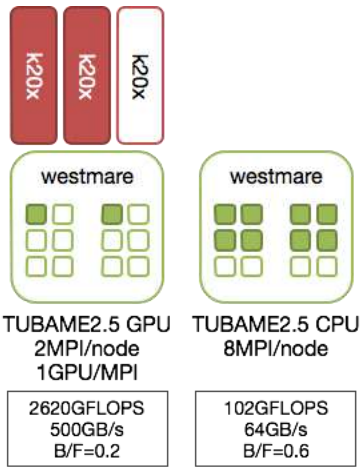

図1：TSUBAME2.5でNICAM-DCを実行した際の1ノードあたりの構成。それぞれGPU使用時（左）、GPU非使用時（右）を示す。

<演算性能>

TSUBAME２．５を用いて、CPU のみでの実行と GPU を利用した実行を行い、演算性能の評価を行った。CPU と GPU の性能比較の方法についてはいくつか方法があるが、NICAM はメモリ律速のアプリケーションであるため、図1に示すノード単位での比較を行った。このとき、CPU のみの実行では AoS 版のカーネルを用い、 MPI プロセス分割数を 4 倍に増やして行った。GPU を用いた実行では TSUBAME2.5 のノードに 3 枚搭載されている GPU カードのうち、2 枚を用いている。ノードあたりのプロセス数、GPU 利用数が限られているのは、NICAM の MPI 分割数に制限があり、$10x4^n$ の公約数しかとれないためである。計測にあたり、演算量およびメモリ転送量をあらかじめ取得しておき、実行時にはタイマー計測で実行時間を取得した。電力消費量については、TSUBAME2.5

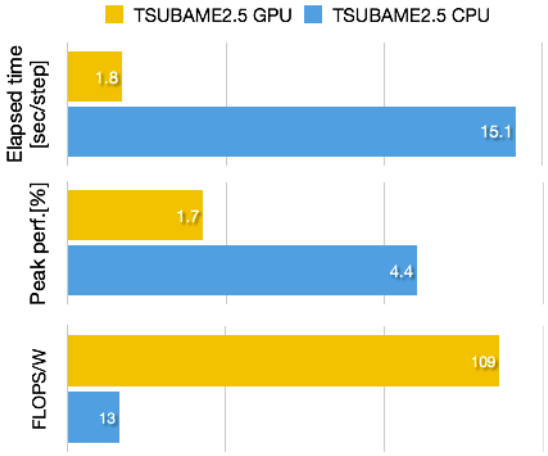

図2：TSUBAME2.5でNICAM-DCを実行した際の実行性能。それぞれメインループ1ステップあたりの所要時間(上)、演算ピーク性能比（中）、ワットあたりFLOPS(下)を示す。

がサービスとして提供するノード単位の電力消費情報のためのツールを利用した。性能評価に用いた際の問題サイズは水平解像度 56km、鉛直 160 層（格子数 2600 万）であった。セットアップを除くメインループの1ステップあたり総演算量は 420GFLOP/step であり、メモリ転送量は 2.3TB/step であった。GPU は5ノード 10MPI プロセス、CPU は5ノード 40MPI プロセスで計測を行った。演算性能評価のためのテストケースには Jablonowski and Williamson（2006）[3]の傾圧不安定波実験(ただし 60 ステップ)を用い、結果取得のためのファイル出力も行った。結果を図２に示す。図より、OpenACC を通して GPU を利用した際の実行時間は CPU のみを利用した時よりも、およそ8倍高速に実行された。これは図1に示した構成での、メモリ転送性能の差とおおむね一致している。CPU、GPU 実行においてどちらもピークメモリ転送性能の約 50%を利用できており、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を示している。一方、演算ピーク性能比という観点では、より Byte/FLOP 比の小さい GPU でピーク性能比が悪い結果となった。NICAM-DC の主要部分である力学コアは NICAM の中でも要求 B/F 比が高い部分であり、強くメモリ転送性能に律速される。GPU の演算性能はまだまだ活用出来る余地が残っており、シミュレーション結果の物理的な性能や演算量を評価しながら、力学コアのスキーム・アルゴリズムの変更や実数演算精度の見直しを行うことが今後必要であると考える。消費電力あたりの演算性能では、実行時間とおおむね同じ程度の差が得られた。

<スケーリング性能>

図3にウィークスケーリング性能の計測結果を示す。ウィークスケーリングではモデルの水平解像度を 56km から

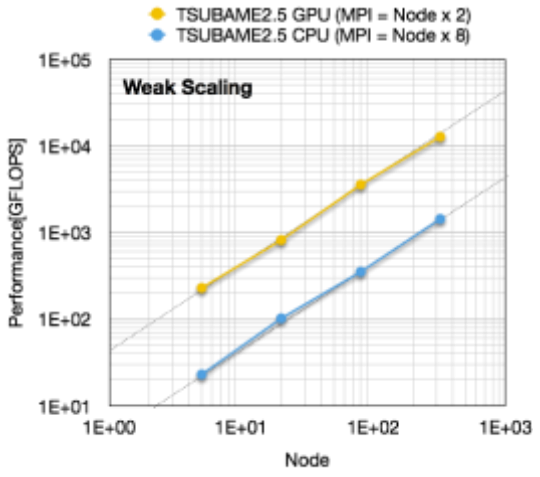

図3：TSUBAME2.5でNICAM-DCを実行した際のウィークスケーリング性能。

ら 7km まで変化させることで、ノードあたりの問題サイズを固定して 5 ノード 10MPI プロセスから 320 ノード 640MPI プロセスまで増加させた。図よりわかるように、CPU、GPU 実験共に良好なスケーリング性能が得られている。1ステップあたりの実行時間でみても、5 ノード実験に対し 320 ノード実験で 30%程度所要時間が伸びるのに留まっている。所要時間が増加する主な要因は、CPU 実行の場合は主にネットワーク通信時間の増加であるのに対し、GPU 実行の場合はヒストリ出力であった。CPU の場合はフラット MPI で実行しているために、ネットワーク通信の輻輳が MPI プロセス数の少ない GPU 実行時より起こりやすいと推察される。一方、GPU ではファイル出力を行う変数の3次元配列全体を GPU からホストに転送する必要があり、袖通信のためのデータ転送と比べても転送量が圧倒的に多く、律速となっていることがわかった。時系列データは多くの場合倍精度である必要がないので、今後は GPU 上で出来るだけ単精度化して転送するよう改良する予定である。

図4にストロングスケーリング性能の評価結果を示す。

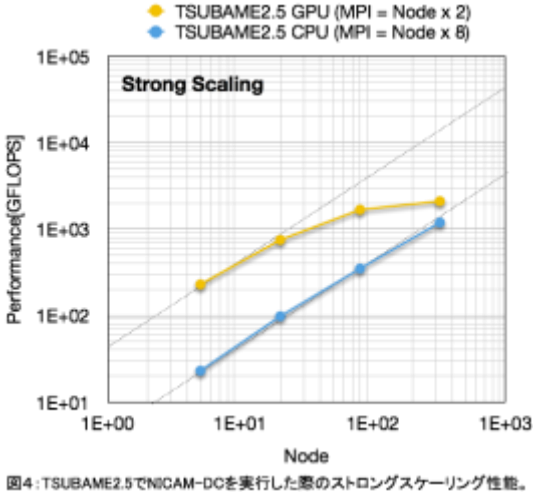

図4：TSUBAME2.5でNICAM-DCを実行した際のストロングスケーリング性能。

ストロングスケーリングでは水平解像度 56km のまま、5ノードから 320 ノードまで増加させた。このときプロセスあたりの水平格子数は 16900grid から 100grid まで減少している。図より、CPU のみの実行と比較して、GPU 実行の場合はノード数の増加に伴って急速にスケーリング性能が低下していることがわかる。これは GPU-Host 間のデータ転送時間の割合が増大することに加えて、水平格子数の減少に伴い GPU 上で実行されるスレッド数が減少し、十分な演算効率を発揮出来なくなったためである。特にデータ転送に関しては、1回の袖通信のために行う一連の動作に含まれる種々のレイテンシが無視出来ないため、通信量よりも通信回数を減らすことが効果的であることがわかった。また、課題実施時点の TSUBAME2.5 ではサポートされていなかった pinned memory 機能の活用も、GPU-Host 間の転送時間削減に有効であると考えられる。ストロングスケール性能は十年～百年(10^6-10^7 ステップ)にわたる気候実験を行う際に重要であり、今後重点的に改良を進める必要があると考える。

まとめ、今後の課題

OpenACC を用いて、全球高解像大気モデルの力学コア全体を GPU 上で計算し、メモリ転送性能に見合う実行性能を得ることに成功した。ソースコードの変更量はモデル全体の行数と比較して5％以下であり、高いポータビリティを同時に達成した。スケーリングについては、ウィークスケーリングで良好な結果を得た一方、ストロングスケーリングでは GPU-Host 間の転送時間がより律速することがわかった。本課題の結果は、OpenACC を用いることで、ソースコード規模が大きく複雑なアプリケーションコードを容易にアクセラレータ上で動作させられることを実証するものである。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、フルパッケージの GPU 対応を進める。力学コアの更なる高速化に向けて、時間発展スキームの変更や計算結果の精度を考慮した単精度・倍精度計算の混合を検討する必要が考えられる。ストロングスケーリング性能の向上については、通信回数の削減を目指したアルゴリズムの見直しを行うことが重要である。

参考文献

[1]Satoh, M., T. Matsuno, H. Tomita, H. Miura, T. Nasuno and

S. Iga (2008) : Nonhydrostatic Icosahedral Atmospheric Model (NICAM) for global cloud resolving simulations. J. Comp. Phys., the special issue on Predicting Weather, Climate and Extreme events, 227, 3486-3514.

[2]Tomita, H. and Satoh, M. (2004) : A new dynamical framework of nonhydrostatic global model using the icosahedral grid. Fluid Dyn. Res., 34, 357-400.

[3] Jablonowski C, Williamson DL. A baroclinic instabilitiy test case for atmospheric model dynamical cores. Quart. J. Roy. Meteor. Soc. 2006; 132(621C):2943–2975.