

TSUBAME 共同利用 平成 26 年度 学術利用 成果報告書

不均質地球構造における地震波・津波伝播シミュレーション
Simulation of seismic/tsunami wave propagation in the heterogeneous earth

古村 孝志^{1,2}・前田 拓人¹
Takashi Furumura and Takuto Maeda

東京大学地震研究所¹・情報学環総合防災情報研究センター²
Earthquake Research Institute¹/Center for Integrated Disaster Information Research²,
The University of Tokyo
<http://www.eri.u-tokyo.ac.jp/people/furumura>
<http://www.eri.u-tokyo.ac.jp/people/maeda>

邦文抄録 (300 字程度)

差分法による並列地震動シミュレーションコード SEISM3D に対し、OpenACC を用いた GPU 計算を実現し、その性能評価を行った。TSUBEME システムにおける複数の実装タイプに対する性能評価の結果、これまで一般に行われてきた CUDA を用いたコードの大幅な書き直しを必要とすることなく、OpenACC のみで最大 2.92 倍の加速を確認した。データ出力部分を除けばウィークスケーリング性能も良好であり、今後の大規模シミュレーションへの適用が期待される。

英文抄録 (100 words 程度)

We implemented OpenACC-based GPU computing in SEISM3D, a parallelized finite-difference-method simulation software of seismic wave propagation. Through performance measurements of different types of implementation on TSUBEME system, x2.92 speed up at maximum has been achieved only by introducing OpenACC and without significant rewriting of code.

Keywords: 地震動, 波動伝播, 差分法, OpenACC

背景と目的

地震学・固体地球物理学においては地震波形記録が地球内部構造ならびに震源での断層破壊過程を知るための基礎的なデータである。日本国内には 1000 点を越す稠密な基盤的連続観測網が敷かれ、日々地震動記録を蓄積している。さらにこの観測網は海域に向かって拡大を続けており、海底における巨大な地震・津波観測網も構築されつつある。大型計算機による地震動シミュレーションは、これら観測量を解析・説明するための基礎的なツールであり、観測波形と計算波形との比較や同化によってより詳細な地球内部構造や震源の破壊過程像が得られるようになることが期待されている。加えて、地震波動は不均質媒質中における波動伝播の物理学そのもの重要な一フィールドでもあり、不均質な地球内部構造中

における地震波動伝播過程そのものも重要な研究対象である。しかし、観測網の充実に比して、日本列島スケールの空間規模において地震動の主たる周波数帯域 (0–15 Hz 程度まで) をカバーするにはまだ計算機能力は十分とは言えず、継続的な地震動シミュレーション手法ならびにコードの継続的な高度化が求められている。

本課題で主として取り扱う地震動シミュレーションソフトウェアは隣接ノード通信を伴う等間隔格子の差分法コード (e.g., Maeda and Furumura, 2013) であり、地球シミュレータ (Furumura and Chen, 2005) や京コンピュータ (Maeda et al., 2013) をはじめとした種々のプラットフォーム上での動作実績を持つ。コードは各アーキテクチャ上でチューニングされており、京コンピュータではピーク

性能比 20% を超えるなど大規模並列計算機上で十分な性能を達成してきた (井上・他, 2013)。しかし、シミュレーションが密媒質中のベクトル弾性波である以上必然的に多くの 3 次元変数を必要とし、かつ 3 次元的に不均質な地球内部構造も記憶参照される必要があるため、アプリケーションは全体とし

GPU 環境双方で統一したコードを用いることが出来るようになる可能性がある。

そこで、本報告では現在利用されている地震動コードの OpenACC 化を試み、OpenACC による GPU を用いた地震動シミュレーションの実現可能性についての検討を行った。また、TSUBAME をはじめと

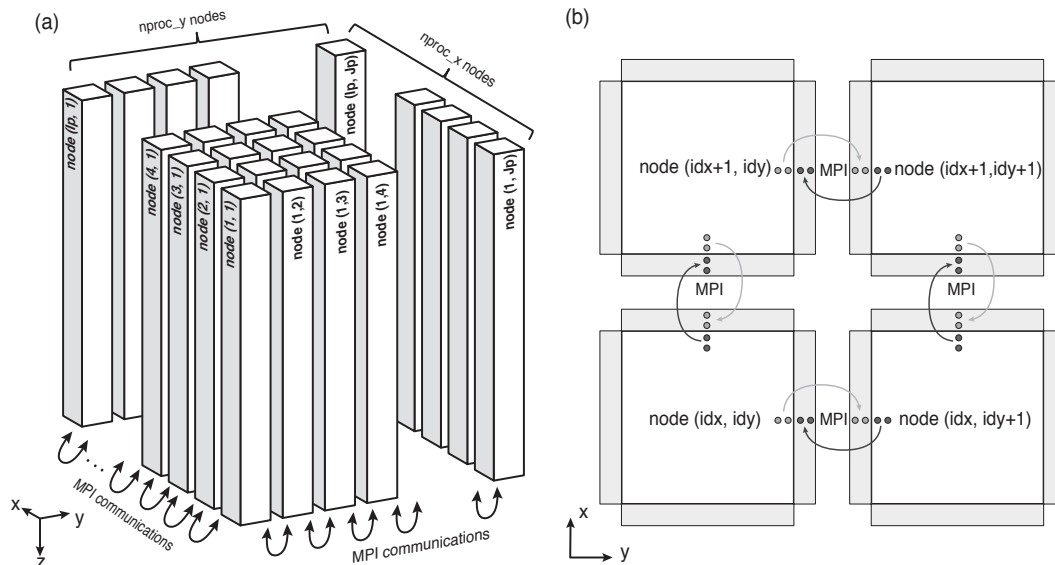


図 1. SEISM3D の並列化レイアウト (Modified from Maeda et al., 2013)

てメモリインテンシブなコードである。今後は計算能力に対するメモリからのデータ供給能力が低下していくと想定されるが、GPU 等の加速器を利用することで大きく効率を改善できる可能性がある。実際に、すでに TSUBAME を含む CUDA 環境を用いて大規模な地震動シミュレーションを実現した例が複数あり (e.g., Okamoto et al., 2010) 地震動シミュレーションへの GPU 利用の有効性が示されている。しかし、アプリケーションコードが CUDA を用いた事実上の書き直しになることが GPU を利用するうえでの大きな困難となっていた。

近年、ハード面では GPU 内蔵のメモリ量が地震動シミュレーションに十分耐えうる程度に増加したこと、ソフト面では OpenACC が標準化され普及されつつあることによって、既存コードを GPU 対応化するための敷居が大きく下がってきた。OpenACC は対応環境以外では無視される指示子によって記述できるため、既存コードからの移植がきわめて容易であると期待されるほか、CPU と

した複数環境における CPU からの性能向上の比較と将来的な大規模 GPU 計算の可能性の検討を行った。

概要

本課題で用いる地震動シミュレーションコード SEISM3D (after Furumura and Chen, 2005) は、地球内部構造を粘弾性体近似のもとに差分法で解く並列コードである。粘弾性緩和は地震動の周波数帯域では地震波エネルギーの散逸 (内部減衰) を表現する。地震動シミュレーションにおいては簡便のため完全弾性体に一定のダンピングを用いることが広く行われてきたが、本コードは特徴的時間の異なる複数の粘弾性体モデルを並列させて扱うことで、広い周波数帯域において地球内部構造の内部減衰特性を良く再現したシミュレーションが実現可能になった [Maeda et al., 2013]。

解くべき運動方程式は、たとえば x 方向について

$$\rho(\mathbf{x}) \frac{\partial v_x(\mathbf{x}, t)}{\partial t} = \frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + \frac{\partial \sigma_{xz}}{\partial z}$$

と表される。ここで ρ は質量密度、 v_x は弾性体変位速度の x 成分、 σ_{ij} は応力テンソルであり、等方粘弾性体では 6 つの独立な成分を持つ。応力テンソルは構成方程式により変位あるいは速度と結びつけられる。粘弾性体の場合には、この構成

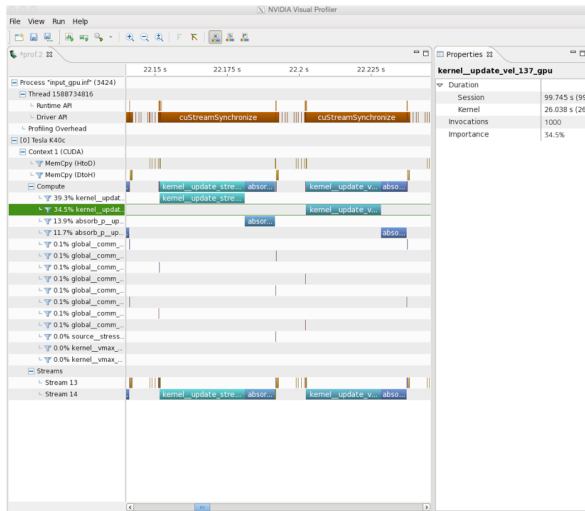


図 2. NVVP によるプロファイル結果の例.

方程式は粘弾性の緩和時間関数時間方向のたたみ込みになるが、そのままでは数値計算上不利なため、メモリ変数(Robertsson, et al., 1994)という手法により、これを複数の時間発展方程式に分割する。本コードで採用されている粘弾性モデル Generalized Zener Body の場合について、 xy 成分は

$$\frac{\partial \sigma_{xy}(\mathbf{x}, t)}{\partial t} = 2\mu_R \left[1 - \frac{1}{N_M} \sum_{m=1}^{N_M} \left(1 - \frac{\tau_{em}^S}{\tau_{om}} \right) \right] \left(\frac{\partial v_x}{\partial y} + \frac{\partial v_y}{\partial x} \right) + \sum_{m=1}^{N_M} r_{xym}(t)$$

$$\frac{\partial r_{xym}(\mathbf{x}, t)}{\partial t} = \frac{1}{N_M} \frac{1}{\tau_{om}} \mu_R \left(1 - \frac{\tau_{em}^S}{\tau_{om}} \right) \left(\frac{\partial v_x}{\partial y} + \frac{\partial v_y}{\partial x} \right)$$

のように表される。ここで、 N_M は並列に用いる粘弾性体モデルの個数、 μ_R は粘弾性体の relaxed moduli で媒質による定数、 τ は粘弾性の緩和時間とクリープ時間で、粘弾性による地震波の内部減衰とその周波数特性を支配する定数である。これらの方程式の右辺を食い違い格子配置の空間 4 次精度の中心差分で評価し、時間 2 次精度の時間積分を行うことで、速度ならびに応力の時間発展問題を交互に解く。

また、計算対象として地球構造の一部を取り出してシミュレーションするため、計算範囲の外周周辺には Perfectly Matched Layer 法 (Chew and Liu, 1996) の吸収境界条件をおいた。これは時間発展方程式を空間微分の方向毎に人工的に分解し、境界面の方向に直交する方向のみ地震波振幅を強減衰させることで、吸収境界からの人工反射波をきわめて小さくする手法である。その一方、内部領域とは異なる時間発展方程式を解き、かつ要求メモリが大きくなるという欠点を持つ。並列化は 3 次元空間を鉛直 2 次元に分割し、分割された領域の境界周辺グリッドを時間積分サイクル毎に隣接ノードと交換することによって行う (図 1)。この計算は多くの変数が 3 次元配列であり演算量の割にメモリ入出力が多く、メモリバンド幅が性能を律速するコードである。

結果および考察

これらの問題を解く既存のコードに OpenACC の指示子を追加することで GPU 用コードの開発を行った。SEISM3D は地球シミュレータや「京」コンピュータ等に代表される様々なアーキテクチャで開発ならびに実行されてきたため、マシンの特徴にあわせた様々な実装がある。そこで、それらの実装の特徴を抽出した 2 種類の実装を OpenACC 化して比較した。

2 種類のコードは時間発展方程式右辺の取り扱いが異なっている。一つ目のコード (コード A) では、独立な 3 重ループで前述の方程式の右辺にある空間微分を評価し、それらを 3 次元配列に一度セーブする。その結果を再度ロードして別の 3 重ループで時間積分を計算する。もう一つ (コード B) では、空間微分の計算と時間積分を融合している。この場合、右辺を格納する 3 次元の一時配列が必要なくなり、利用メモリ量が減る。このことによりメモリ空間の小さなクラスタマシンでも比較的大規模なシミュレーションが実現可能であり、かつ Intel CPU での実行効率はコード A よりも高い。しかし、メモリ削減の代償としてループ内の演算が非常に複雑となり、レジスタ溢

れを起こしやすい。結果的に「京」コンピュータや地球シミュレータ2においてはコードAよりも性能が落ちるという特徴がある。

OpenACC 化は Fortran90 コードのループひとつひとつについて OpenACC の指示子を追加することで行った。本コードで必要な通信は入出力をのぞけば隣接ノード間のみであり、ほとんどのデータは CUDA 上に保持したまま、通信に必要な袖領域

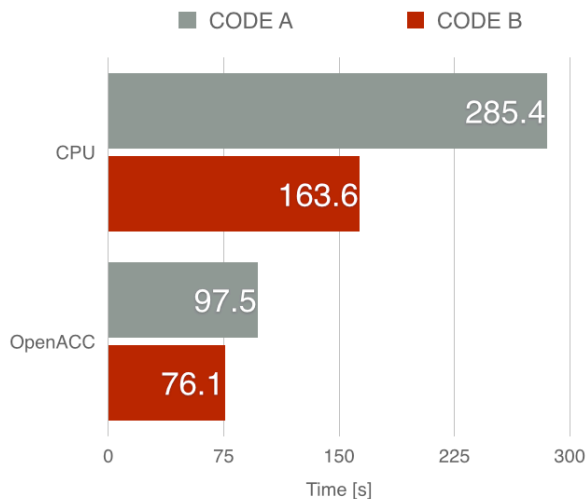


図 3. CPU 実行および OpenACC 利用の GPU 実行の計算時間の比較。

のみを CPU に転送して MPI 通信を行うことができた。このことにより、GPU とホストとの間の通信を最小化し、効率的な計算が可能になった。GPU-CPU 間の通信が MPI 通信およびデータ出力のタイミングでのみ発生し、その他の時間帯は GPU のみで計算が進んでいることを NVIDIA Visual Profiler から確認した (図 2)。

ただし、OpenACC 化されたコードは指示子を除いて完全に CPU 版と同一にすることはできなかった。MPI 通信データのパッキングに用いていた Fortran90 の一部の配列関数が OpenACC コンパイラで非対応であり、該当部分の全面的な書き換えが必要であったほか、性能向上のため、配列のインデックスの順序変更、一部変数の事前計算と 3 次元配列へのストアが必要であった。特に後者のチューニングは CPU 環境での高効率計算と必ずしも整合せず、OpenACC 利用の利点であるコードのポー

タビリティには課題を残す結果となった。

計算実行性能の比較のため、まず最初に TSBAME の 2 ノードを用いた試験を行った。CPU での実行は 4CPU24 コアを Flat-MPI で用い、GPU の場合には 4GPU 一つ一つに CPU の 1 コアを割り当てた。モデルサイズは両者共通に 384x256x256 グリッドを用い、1000 時間ステップの計算を行った。両者の計算実行時間の比較を図 3 に示す。

いずれも CPU に比べて相応の性能向上が確認された。しかし、性能向上率には実装の仕方による顕著な差異が見られた。3 次元一時配列を用いたコード A では 2.92 倍の加速率を達成したのに比べ、省メモリのコード B では 2.14 版にとどまった。コード B ではループが融合し複雑化した結果、利用レジスタ数が多くなったため加速率に限界が表れたものと考えられる。コード B は当初はより加速率低い状態であったが、コンパイルオプションの調整による最大レジスタサイズの指定、再内レベルでのループの分割、OpenACC の vector 指示子のサイズ調整などのチューニングを施した結果、2 倍を超える加速率を達成した。

次に、コード B を対象として、GPU 計算のウィークスケーリングテストを実施した。ノードあたりのグリッド数を 64x64x512、計算時間ステップを 1000 ステップに固定し、3GPU から 384GPU までの評価を実施した。結果を図 4 に示す。

出力がない場合スケーリング性能は良好であり、GPU が特に少ない場合ではスケーリングからのずれが見られるが、これは吸収境界条件が外周部周辺で常に一定のグリッド数を占めるため、特に問題規模が小さすぎる場合には計算量がノード数と完全には比例しないためと考えられる。それ以上、24GPU 以上の場合にはほぼ一定の計算時間で計算規模を拡大することができている。しかし、データの出力がスケーリング性能に顕著な影響を及ぼすことも明らかになった。

SEISM3D は一定時間ステップ毎に、3 次元空間中の 2 次元断面の変数列をバイナリデータとしてファイル出力している。出力されるデータは複数の MPI ノードにまたがるため、MPI の集合通信を用い

てこれを特定ノードにまず集約した上でファイルを出力する。GPU 計算の場合には、まず出力データバッファを GPU から CPU に転送し、さらに CPU 間で MPI 通信がなされるため、通信時間が大きなオーバーヘッドになっている物と考えられる。これは今後 MPI 通信ならびにデータ出力の GPU 計算とのオーバーラップすることによる隠蔽を実装することで大きく改善する可能性がある。

まとめ、今後の課題

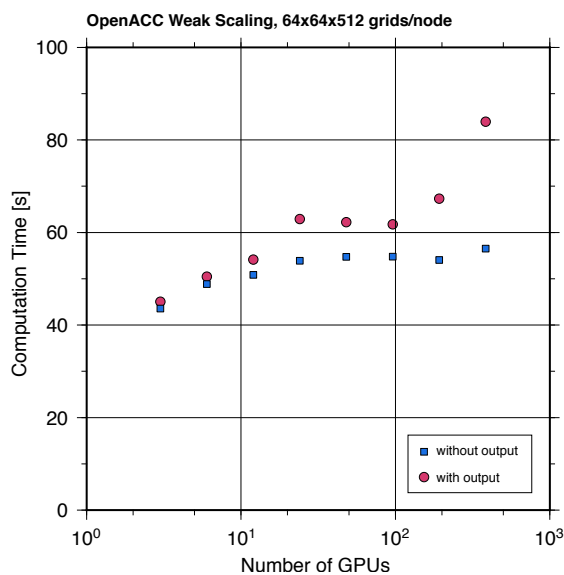


図 4. ウィークスケーリングテストの結果。青四角がデータ出力無し、赤丸がデータ出力ありの場合の計算時間をそれぞれ示す。

地震動シミュレーションコード SEISM3D を OpenACC を用いた GPU 化を行った。CPU と GPU で完全に同一コードとすることまでは実現できなかったが、従来一般に行われてきたような CUDA を用いた大幅な書き直しをすることなく、OpenACC のみで最大 2.92 倍のコード加速を実現できた。多少の性能の差異はあるが、OpenACC 化されたコードはそのまま他の CPU 環境でも実行可能であり、さまざまな計算資源を有効に活用したシミュレーションとそれを用いた固体地球物理学の研究への活用が期待される。また、ウィークスケーリング性能も良好であり、今後の大規模シミュレーションへの適用が十分に可能であると考えられる。ただし、そのためには今後出力

時間の隠蔽を行う必要がある。また、MPI 通信についても隠蔽を行うことでさらなる高速化が可能となるであろう。本報告ではもっぱら地震動シミュレーションを対象にしたが、津波計算の基礎方程式も地震動シミュレーションと似た構造をしており、同様の OpenACC 化を今後行うことでさらなる加速が期待される。

謝辞

OpenACC 化に当たり、NVIDIA 社の成瀬彰氏に貴重なアドバイスをいただきました。記して謝意を表します。