TSUBAME 共同利用　平成 27 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
Takashi Furumura

東京大学地震研究所
Earthquake Research Institute, the University of Tokyo
URL

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，その性能評価と高度化を行った．OpenACC を用いることで， CUDA を用いたコードの大幅な書き直しをせずに高性能計算が可能である．データ出力については，単独 GPU 上ではデータ出力に伴う GPU-CPU 間の通信の隠蔽に成功したものの，本システム上では MPI による集合通信に時間がかかり，大規模並列計算を困難にしていることが課題として残された．

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. We implemented concealed data communication between GPU and CPU for snapshot data output, however, the collective data communication among CPUs still takes time, that makes large-scale computation difficult.

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と期待されている．さらに，地震波動は不均質媒質中における波動伝播の物理学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，観測される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能力は十分とは言えず，継続的な大規模並列地震動シミュレーション手法ならびにコードの高精度化に向けた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（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ホーム上での動作実績を持つ．本質的に多変数である地球内部不均質構造や，テンソルならびにベクトル量である地震動と応力状態を 3 次元的に維持する必要から，本コードは多量のメモリ容量を要し，かつ計算量に比してメモリアクセス量の多いものである．そのため，コードの性能は主としてメモリバンド幅によって律速される（井上・他, 2012）.

長い間用いられてきた CPU を中心とした機構に加え， GPU でも地震動を計算できる可能性がこれまでに示されてきた．本課題で扱う計算コードは，

研究の進展に応じて頻繁に改変や改良が非常に頻繁に施されるものである．そのため，アーキテクチャ間で共通のコード構造を維持しつつ，都度最適な計算資源を利用できることが実用上きわめて重要である．その目的に対し，OpenACC はコードのポータビリティを維持しつつ GPU の比較的豊富なメモリバンド幅を活用できるという点で優れており，地震動シミュレーションや，それと非常に近いアルゴリズムで解かれる津波のシミュレーションの高度化と活用に大きく貢献する可能性がある．

これまでの予備研究から，既存コードに OpenACC の指示子を追加することで，GPU で CPU より高い性能を出し得ることが示されてきた．しかし，性能評価からはデータ出力や隣接ノード間通信が計算を律速しており，この部分を改善することで複数 GPU ボードを用いた実用的な大規模計算が可能になると期待される．

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．解くべき運動方程式はたとえばx方向について

$$\rho \frac{\partial v_x(\boldsymbol{x},t)}{\partial t} = \frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + \frac{\partial \sigma_{xz}}{\partial z} \quad (1)$$

と表される．ここでv_xは弾性体変位速度，ρは密度，σ_{ij}は応力テンソルの成分であり，3 次元媒質では独立な成分は 6 つある．右辺は均等に配置した食い違い格子上の値を差分法によって評価する．これと応力についての構成方程式の時間微分ならびにその表現に必要なメモリ変数の時間発展方程式

$$\frac{\partial \sigma_{xy}(\boldsymbol{x},t)}{\partial t} = 2\mu_R^* \left(\frac{\partial v_x}{\partial y} + \frac{\partial v_y}{\partial x} \right) + \sum_{m=1}^{N_m} r_{xym}(t) \quad (2)$$

$$\frac{\partial r_{xym}(\boldsymbol{x},t)}{\partial t} = \frac{\mu_R}{N_m \tau_{\sigma m}} \left(1 - \frac{\tau_{\epsilon m}^s}{\tau_{\sigma m}} \right) \left(\frac{\partial v_x}{\partial y} + \frac{\partial v_y}{\partial x} \right) \quad (3)$$

を交互に時間積分することで問題を解く．運動方程式と構成方程式の積分を行うたびに，MPI により袖領域の情報を隣接ノード間通信により交換し，媒質の連続性を担保する．本コードでは 3 次元空間を水平 2 次元に分割するため，隣接ノード通信は計 4 方向に対して行う．GPU 計算においては，OpenACC の指示子により通信に必要な部分のデータを CPU に送信，その後 MPI で通信を行い，受信を完了した時点でふたたび OpenACC 指示子により GPU にデータを戻していた．CPU のみでの計算に比べ，CPU と GPU の間の通信が必要となる分通信時間が増え，また CPU 間の通信の間 GPU は待ち状態になる問題があった．

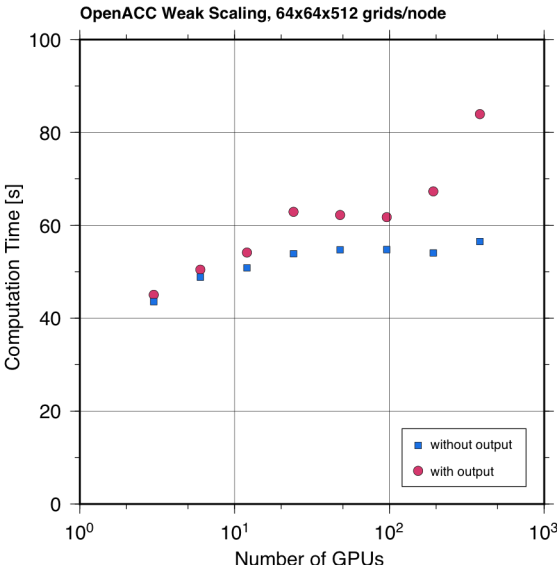

図 1. データ出力あり（赤）となし（青）の場合の計算実行時間の比較（平成 26 年度報告書より再掲）

データの出力は，地震計の出力を摸した地表面の特定の点における 3 成分の変位速度波形と変位波形記録と，波動場全体を評価するための時空間スナップショットの二つについて行われる．このうち前者は負荷が無視できるほど小さいことがわかっているため，評価から除外した．後者のスナップショットは，事前に指定した xy, xz, xy の 3 つの平面と海表面ならびに海底面の，合計 5 つの 2 次元断面における地震波およびその空間微分を出力するものである．ポストプロセッシングの単純化のため，各ノードのデータを単一ノードに集約し，断面あたり一つのバイナリファイルとして出力する．GPU 計算においては，隣接ノード間通信と同様に，GPU 各ノード上で出力する変数をバッファリングし，GPU から CPU に送信，それをさらに CPU 間で集約通信を行い，IO を担当するノードからファイルを出力する．

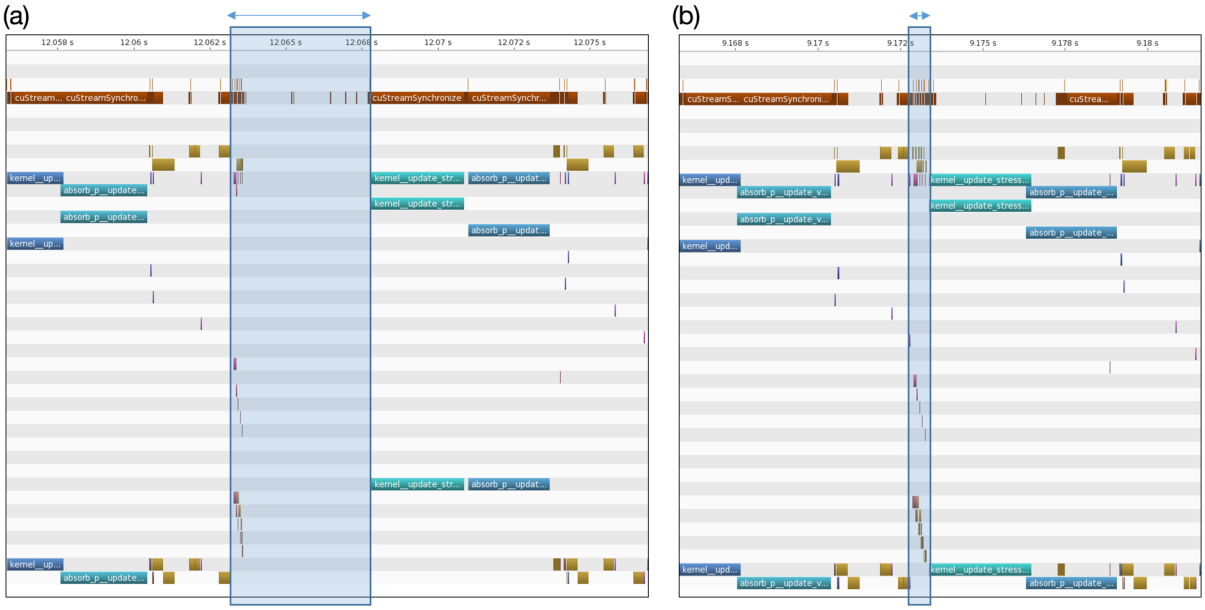

図 2. スナップショット出力の GPU-CPU 通信の隠蔽の効果．(a)チューニング前，(b)チューニング後のデータ出力を含む計算 1 時間サイクル分の NVIDIA Visual Profiler の出力．それぞれ網掛部分が出力データのバッファリングと CPU への通信ならびに出力部分に該当する．

TSUBAME においては，平成 26 年度の検討において，特にこのデータ出力の負荷がノード数増大に伴い増大し，大規模な計算を困難にしていることがわかっている（図 1）．

結果および考察

まず，TSUBAME 上においてデータ出力の時間増の原因を探るため，16 ノード 48GPU を用いた計算において比較を行った．まず，スナップショットを全て出力する高負荷の状況において，オリジナルコードの計算時間を測定したところ，62.3 秒であった．MPI 通信の影響を抽出するため，単一ノードへの集合通信をやめて，各ノードから分割ファイルを出力するように変更すると，計算時間は 55.4 秒に短縮された．これは出力なしにした場合の計算時間（図 1）と同程度であり，したがって TSUBAME における計算時間の増大は集合通信（mpi_reduce）の影響が支配的であることが確認された．そのため，データ出力の方式を大幅に変える以外には現行の TSUBAME システムで大容量出力をともなう大規模シミュレーションを実行するのは困難であることがわかった．今後，他のアーキテクチャとの可搬性も考えながら，CPU レベルでの集合入出力（MPI-IO）やそれを用いたライブラリの利用を検討する必要がある．

一方，上記検討とは独立に，今後の GPU 利用に向けたデータ出力のための GPU-CPU 間通信の最適化についても検討した．従来コードは(1)式の評価後に出力データのバッファリングからファイル出力までをまとめて行っていた．これを，(1)の評価の直後には GPU 上でのバッファリングと CPU への送信(A1)だけを行い，その後に(2), (3)の応力評価(B)とそれに付随する境界条件処理(C)を行い，その後にバッファされたデータの出力(A2)を行うようにコードを改良した．加えて，OpenACC の async 節とその引数を用いて．(A1), (B), (C)の処理が非同期に，かつ(A1)の処理終了後に(A2)が順番に，それぞれ実行されるようにした．また，複数の出力種別のバッファリングも非同期に行われるよう設定した．このことにより，バッファリングされたデータの CPU への送信と応力評価が違いにオーバーラップして実行されるようになり（図 2），単独 GPU においては処理時間の大幅な短縮が確認された．また，内部領域と吸収境界条件の計算についても多少ではあるが

非同期実行が行われ，適切な async 節の利用が計算時間の短縮に寄与することが確認された．

今後, CPU からファイル出力の部分が改善される，あるいは次期システム等で集約通信に時間のかかる問題が軽減されれば，本最適化によってさらなる高性能が獲得できると期待される．

まとめ、今後の課題

地震動シミュレーションコード SEISM3D の OpenACC を用いた GPU 版について，特にデータ出力の高度化のための検討を行った．OpenACC を用いたことで，少量のコード書き換えで，CPU 計算との互換性をほぼ維持しつつ計算の高速化を達成したが，一方本アーキテクチャにおいては集合通信に時間がかかることがわかった．大量のデータ出力を含む実用計算は，データ出力の分散化を含む大幅な書き換えを行わない限り困難であると考えられる．一方，単独 GPU における検討からは，データ出力のためのバッファリング結果を CPU に送信する時間を，計算とのオーバーラップによって非常に効果的に隠蔽できる可能性が示された．今後のデータ出力部分の改善もしくはアーキテクチャの変更により，ポータビリティを維持したままに GPU を活用した大規模計算が可能になると期待される．