

6

「京」コンピュータによる大規模津波
シミュレーション

—津波伝搬遡上コードの超高並列向け最適化—

安藤和人¹ 馬場俊孝¹ 松岡大祐¹ 加藤季広²

1 海洋研究開発機構 2 NEC

東日本大震災のような大規模津波災害の場合、救助や救援物資の手配といった緊急災害対応のために、広域にまたがる被災の状況を詳細かつ迅速に把握しなければならない。しかし、航空機を用いた浸水域調査では調査範囲が限られ、また衛星画像についても、衛星が被災地上空を通過しなければ取得できない。ましてや、夜間に津波災害が発生した場合、被災状況の把握は夜が明けるまでほぼ不可能である。このため、津波警報に間に合う早さで予測計算を完了できれば被害軽減に資するのはもちろんであるが、たとえ間に合わなくとも、可能なかぎり高速で広域かつ詳細な浸水予測を行う必要がある。なぜなら、救援活動の実施主体となる各自治体の緊急対策本部は、まず救援活動の対象地域・分量・種別といった重要な決定を下す必要があるが、その自治体に対して迅速かつ適切な救助活動と人的被害の軽減に資する情報を提供することが可能となるためである。本稿ではこれを目的として開発した超高並列向け津波伝播遡上コード (JAGURS) とその最適化について紹介する。この最適化により、「京」における1万ノード規模での実行において、オリジナルの MPI_Alltoallv を使用した実装に比べ19.4倍、3次元Tofu専用アルゴリズム適用時の MPI_Allreduce を使用した実装に比べ2.2倍の高速化を実現した。これにより、高知県全域を5m分解能とした広域詳細モデル (3層ネスティング、計6.8億格子) では、5時間分の津波伝搬計算を1.5時間で終了することができた。大規模津波災害発生後の緊急災害対応の検討資料として間に合う時間を考慮し、当初3時間での実行完了を目標としていた。1.5時間という結果は、それと比べて十分なスピードと言える。

アプリケーション概要

● シミュレーション内容

地震は地殻変動を伴い、地盤の水平移動や昇降が発生する。これが海底において生じた場合、海底の昇降によって津波が励起される。JAGURSは津波の波源から沿岸までの伝搬、および陸地への浸水や河川への遡上を計算することができる (図-1)。震源情報を元に作成した波源を初期条件、陸域と海域を含む地形情報を境界条件として入力し、各地点での波高と流速の時間発展を評価することにより、ある波源を与えた時の各エリアの浸水被害状況をシミュレーションする。

● スキーム

長波方程式の差分法

津波は「長波」と呼ばれる非常に波長の長い波である。JAGURSでは流体方程式 (Navier-Stokes方程式) を長波として近似した式 (長波方程式) を

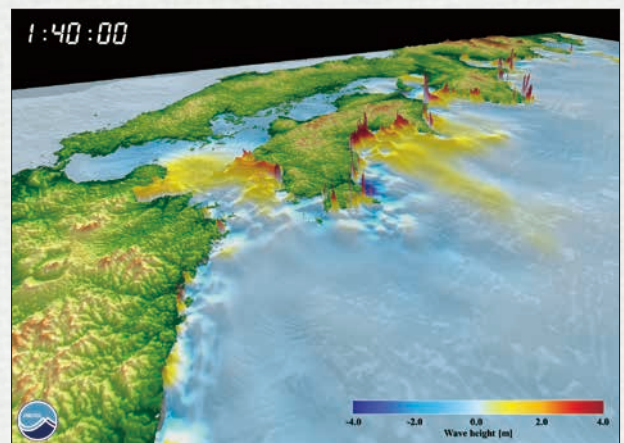


図-1 JAGURSによる南海トラフ地震の津波計算

スーパーコンピュータ「京」の利用

使用し、その時間発展を有限差分法により解いている。移流項（空間の移動に起因する速度の変化量）および海底面の摩擦を考慮した非線形式、またはこれらを考慮しない線形式の2つに対応。また太平洋を伝搬するような津波計算も扱えるように座標系は球面座標が採用されている。以下に JAGURS で用いている支配方程式¹⁾を示す。

運動方程式

$$\frac{\partial u}{\partial t} + \frac{u}{R \sin \theta} \frac{\partial u}{\partial \phi} + \frac{v}{R} \frac{\partial u}{\partial \theta} = -\frac{g}{R \sin \theta} \frac{\partial h}{\partial \phi} - C_f \frac{u \sqrt{u^2 + v^2}}{d+h}$$

$$\frac{\partial v}{\partial t} + \frac{u}{R \sin \theta} \frac{\partial v}{\partial \phi} + \frac{v}{R} \frac{\partial v}{\partial \theta} = -\frac{g}{R} \frac{\partial h}{\partial \theta} - C_f \frac{v \sqrt{u^2 + v^2}}{d+h}$$

連続の式

$$\frac{\partial(d+h)}{\partial t} = -\frac{1}{R \sin \theta} \left[\frac{\partial\{u(d+h)\}}{\partial \phi} + \frac{\partial\{v \sin \theta(d+h)\}}{\partial \theta} \right]$$

ここで、 t は時刻、 R は地球半径、 ϕ は経度、 θ は余緯度、 u 、 v はそれぞれ経度方向、緯度方向の流速、 d は水深、 g は重力加速度、 C_f は摩擦係数、 h は波高である。差分スキームは、スタaggerド格子^{☆1}を採用し、時間方向はリープフロッグ法^{☆2}を使用している。数値不安定を避けるため移流項のみ1次の風上差分^{☆3}で解いている。

ネスティング計算

JAGURS は、計算領域内の特定の領域について格子解像度を上げた計算（ネスティングアルゴリズム）が可能である。多階層のネスティング（図-2）および複数の子領域の設定も可能であり、またシームレスな計算を実現するためネスティング階層間では双方向の通信が行われる。親領域から子領域へ受け渡すデータは親のデータを内挿して生成し、子領

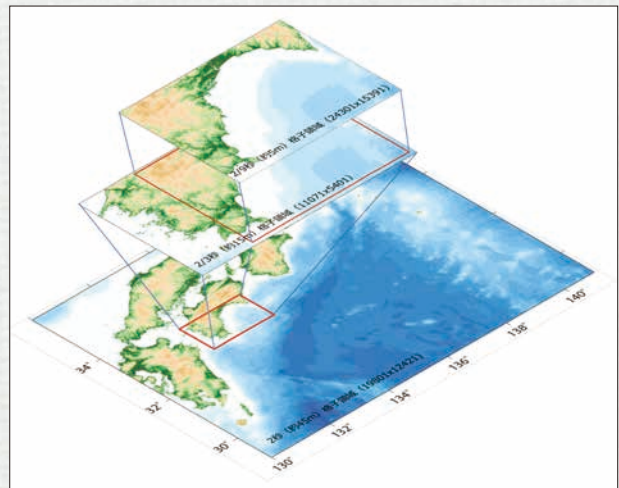


図-2 地形ネスティングの例

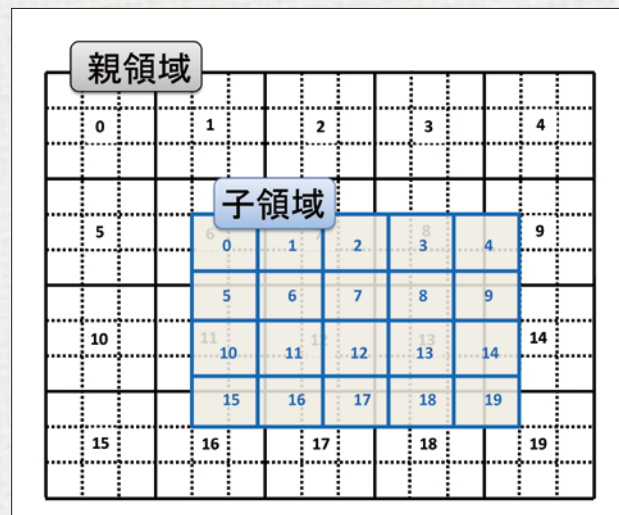


図-3 MPI 領域分割とネスティングの関係。番号は、矩形で囲われた領域の計算を受け持つランク番号を示す。

域から親領域へは、対応する子のデータをそのままコピーしている。

並列化

コードには MPI^{☆4}と OpenMP (+ コンパイラによる自動並列化) を使用したハイブリッド並列化が施されている。MPI 並列化については、ネスティング階層それぞれについて、全プロセスでの2次元領域分割を行う（図-3）、グリッド内の分割領域間の通信は1対1の袖通信、またネスティング階層間の通信は全プロセスによる集団通信となる。一方 OpenMP によるスレッド並列化については、主に

☆1 更新計算時に相互に参照する変数同士を半格子ずれた位置に定義する方法。計算量やメモリ使用量を抑えたまま、半分の格子幅相当の計算精度を得ることが可能である。

☆2 空間微分項の差分化において参照するタイムステップに対し、時間微分項の差分化においてその前後のタイムステップを参照する方法。計算が安定するための条件を緩和することができる。

☆3 上流の性質が下流に影響するという物理的な性質に基づき、流れの方向に応じて上流側の差分格子を参照する方法。移流項の様な非線形項の差分計算における数値安定性を向上することが可能である。

☆4 Message-Passing Interface：分散メモリマシン上での並列計算において使用する通信のためのインタフェースを定義した仕様。

各ランクが担当する分割領域中のグリッドの更新計算をコア間で分担して実行している。

● 従来の津波シミュレーション

同様の計算（ネスティングを使用した長波方程式の差分法）を行うコードとしては、MOST²⁾、TS UNAMI-N2³⁾、COMCOT⁴⁾ 等があるが、多くはプロセス並列化がなされておらず、大規模並列環境向けのコードが十分に用意されているとは言いがたい。他方で離散化のスキームが異なる（非構造格子の有限体積法）ものであれば、「京」向けに最適化がなされた例がある⁵⁾。当該のアプリは現状で4階層のネスティングによる最高解像度15mまで（総セル数1,500万）の計算を実施しており、2,000ノードでのストロングスケーリングが42.2%となっている。

「京」向け最適化

● 対象モデル

南海トラフ地震による高知県の津波浸水計算を対象とし、「京」向け最適化を行った。このモデルのネスティング階層は南海トラフエリア（関東南部～九州東部）全域1,000km×780kmをカバーする50mの格子から、高知県の沿岸部全体を含む5mの格子までの3階層で、総格子数は6.8億である（以下、

高知モデル）。最適化のゴールとしては、災害発生から浸水予測完了までの所要時間としての適切さを考慮し、「5時間分の計算を3時間以内で行うこと」と設定した。

● 単体性能改善

オリジナル版プログラムについて区間ごとの経過時間を測定したところ、最も演算コストの高い個所は、各格子点に対して流速を更新していく2重ループの個所であった（メインループの70%以上のコスト）。このループはネストされたものも含め多数のIF分岐を含むループであり、浮動小数点演算待ちが非常に多かった（当該ループ全体の経過時間の63%ほど）。この部分は、コンパイラによるSIMD化・ソフトウェアパイプライン化がまったくなされていなかったが、処理が高コストとなる分岐を別個のループとして分離し、さらにそれをループ分割することにより、SIMD化・ソフトウェアパイプライン化が促進され、浮動小数点演算待ちは1/11に削減された。結果としてループ全体の経過時間は1/2となり、ピーク性能比としては17.89%まで向上した（オリジナル版は7.34%）（図-4）。

● 並列性能改善

オリジナル版の実装と「京」での性能

ネスト間通信において、親領域の各分割領域を担当するプロセスは、自身の分割領域に一部もしくは全体が含まれる子の分割領域を担当するプロセスに対し、自身のグリッドが保有する流速および波高の情報を転送する必要がある（同様に子から親への通信も行う）。実効的には、子領域の境界部に対応する自身のグリッドのデータを子領域の各担当プロセスに転送し、子領域側で細かいグリッドに内挿補間する形となる。MPI並列数の増加とともに、プログラム全体の実行時間は、上記通信部のコストにより支配されるようになる。

ネスト間通信の実現方法として、オリジナル版ではMPI_Allreduceを使用したものとMPI_Alltoallvを使用したものの2種類を試作し、比較した。

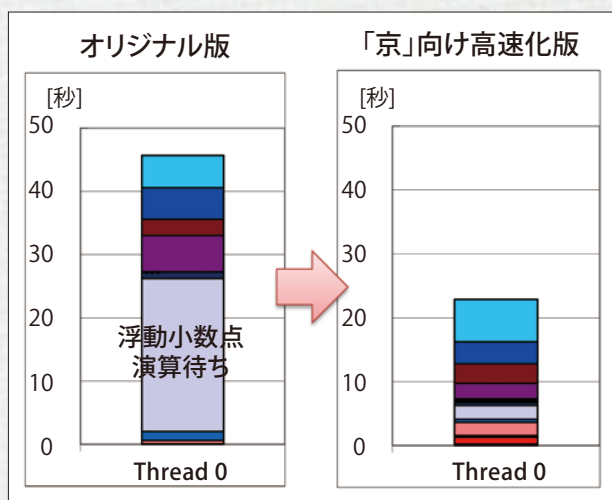


図-4 「京」向け最適化による高コストループの性能改善

スーパーコンピュータ「京」の利用

MPI_Allreduce を使用した実装 (図-5) については、全プロセスが子領域の境界領域の値を計算するのに必要な全要素分の長さを持つ通信バッファを保持し、これを0で初期化する。次に各プロセスが、親領域において自身が担当する分割領域についてのみ、子領域への伝達が必要なデータ要素を通信バッファにコピーする。その後、全プロセスでのMPI_Allreduce 通信を実行することで、全プロセスが、子領域の境界部の値を計算するのに必要な全要素のデータを保持した状態となる。各プロセスは、このデータのうち、子領域において自身が担当する分割領域の計算に必要な部分のみを参照し、境界部の値を計算する。

この手法は、実装が容易であり、かつTofuの3次元論理トラス形状を利用した高度な最適化ルーチンを利用できるメリットがある。一方でデメリットとしては、並列数によらず同サイズの通信バッファを確保する必要がある、各ランクについて、自身の担当とは関係の無い要素まで通信する必要が生じ、通信量の無駄が発生する。

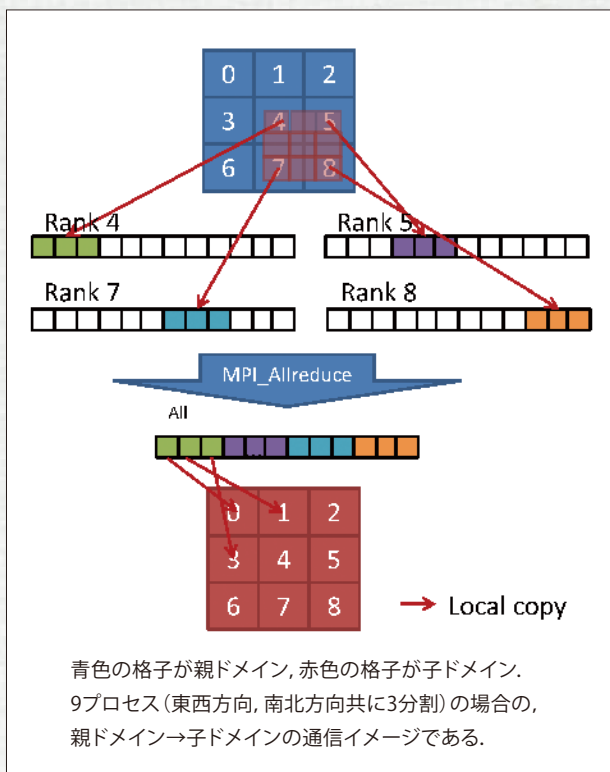


図-5 MPI_Allreduce によるネスト間通信の実装

MPI_Alltoallv を使用した実装については、自身の使用する最低限のバッファを確保し、最低限の通信を行う。メリットとしては、並列数の増加に伴ってプロセスあたりの通信量が減少する点である。一方でデメリットとしては、まず通信部分のアプリへの実装が複雑であること。また MPI_Alltoallv についても Tofu 向けの最適化アルゴリズムが使用可能だが、論理トラス形状に合わせた最適化アルゴリズムは実装されていない点が挙げられる。

上記の2つの実装について、前述の高知モデルを用いて8,748 ノードまでのストロングスケーリングを測定したところ、無駄な通信量が含まれるはずの MPI_Allreduce を使用した実装の方が9倍弱性能が高いという結果となった。

こういった結果となる理由として、MPI_Allreduce については Tofu の論理形状 (3次元トラス) を意識した組込みのアルゴリズムが適用される一方、MPI_Alltoallv については3次元形状に特化した組込みのアルゴリズムが用意されていないためと考えられる。また、今回のケースでは通信データが境界領域に関するもののみであり、データ通信量がさほど大きくない点も、MPI_Allreduce を使用した実装の性能が高かった理由として挙げられる。

3次元論理トラス形状に特化した通信アルゴリズムの最適化

上記の結果を踏まえ、組込みの3次元アルゴリズムが用意されていない MPI_Alltoallv について、全プロセスでの MPI_Alltoallv 通信を、最大3次元の論理トラス形状における各方向での MPI_Alltoallv 通信に分割して行うよう修正した。これにより、プロセスあたりの通信回数のオーダーは (プロセス数を n^3 として) $O(n^3)$ から $3 \cdot O(n)$ に削減され、また通信の衝突を防ぐことも可能となった。

上記の改良に加えて、さらに Tofu のバンド幅を活かすために複数の通信をひとつにまとめた結果、8,748 ノードでの経過時間についてはオリジナルの MPI_Alltoallv を使用した実装に比べて19.4倍、MPI_Allreduce を使用した実装 (3次元 Tofu 専用アルゴリズム適用) に比べ2.2倍の高速化が得られ

るといった結果となった。図-6に各最適化項目によるノード数と経過時間の関係を示す。

ストロングスケール性能を測定した結果、96ノードでの実行時間を基準とした2,520ノードでの並列化効率は53%であった。ネスト間通信を含め、さらなる最適化の余地が残っており、今後もさらなる高並列化・高性能化の実現を進めていく。

● 最適化の意義

最適化の対象とした高知モデルについては、「京」への移植当初、5時間の津波を計算するために7時間の実行時間を要していた。一方、上述の「京」向けの最適化などを施した結果、当初設定していた3時

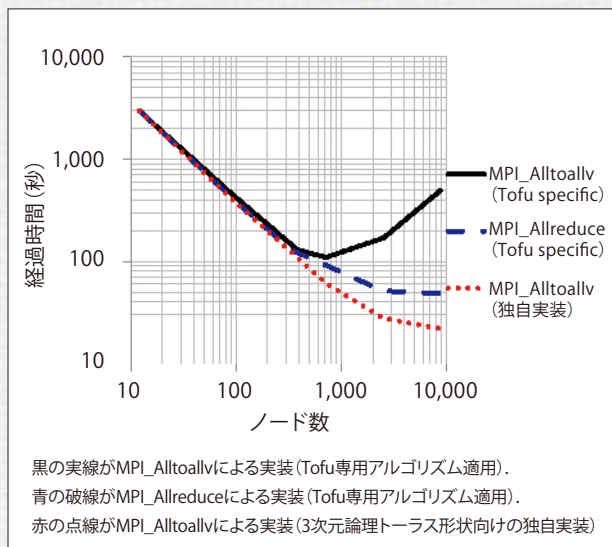


図-6 ネスト間通信方法の違いによるスケール性能の比較

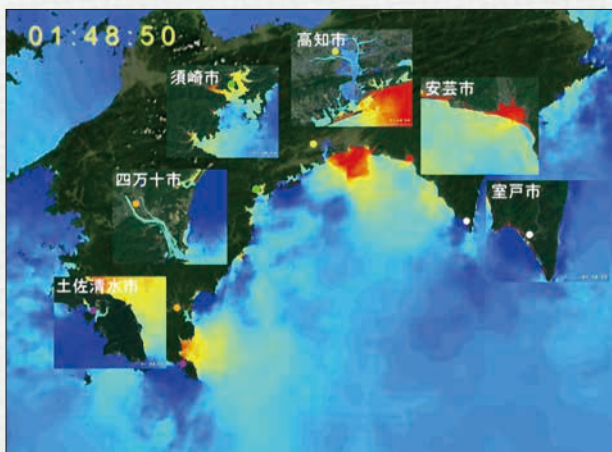


図-7 高知モデルの津波伝搬・浸水計算

間以内での実行という目標を大幅に上回り、1時間半弱での実行が可能となった。

東日本大震災を例にとると、災害発生後1時間半という時間帯は各自治体において災害対策本部が立ち上がり、第1回目の会議が開かれている頃である。TVのヘリ中継など断片的な情報は得られていたものの、この時間帯で日本中の誰ひとりとして災害の全体像を把握できているものはいなかった。このタイミングで高精度、高分解能な津波浸水計算を完了できることは、迅速で的確な緊急災害対応体制の構築に繋がると考えられる(図-7)。

もちろん、実際に発生した津波の波源をどのように高精度かつリアルタイムで推定するか、また巨大な出力データのポスト処理をどのように行うかなどの課題はあるものの、1時間半で広域の詳細な計算を終えられることは、防災上きわめて重要な成果と言える。

さらなる高度化について

津波とそれに伴う現象を詳細に再現し、津波被害の軽減に資するためには、長波理論よりもさらに高度なモデル、具体的には津波の分散性(津波先端部が波長の短い波に分裂する現象)を扱うモデルや津波に伴った土砂の移動を扱うモデル、また津波と漂流物との連成モデルが必要となる。今後はこれらの解析モデルの開発を行うとともに、今後登場するであろうさらなる大規模の計算機へ向けた最適化も進めていく。

なお、本研究は文部科学省のハイパフォーマンス・コンピューティング・インフラ(HPCI)戦略プログラムにおける分野3「防災・減災に資する地球変動予測」において進められたものである。分野3内の各課題は相互の連携を進めており、我々のJAGURSと地震のプログラムの連成計算など、防災計画の多角的な検討に資するプラットフォーム作りを進めていく予定である。

参考文献

- 1) Satake, K. : Linear and Nonlinear Computations of the 1992 Nicaragua Earthquake Tsunami, Pure Appl. Geophys, Vol.144, pp.455-470 (1995).
- 2) Titov, V. V. and Gonzalez, F. I. : Implementation and Testing of the Method of Splitting Tsunami (MOST) Model, Tech. Memo. ERL PMEL-112, Gov. Print. Off., Seattle, WA, p.11 (1997).
- 3) Imamura, F. : Simulation of Wave-Packet Propagation along Sloping Beach by TSUNAMI-Code, in: Long-Wave Runup Models, edited by : Yeh, H., Liu, P. L. E. and Synolakis, C., World Scientific, pp.231-241 (1996).
- 4) Liu, P. L. F., Woo, S. B. and Cho, Y. S. : Computer Program for Tsunami Propagation and Inundation, Tech. Memo, Cornell University, p.104 (1998).
- 5) 今村文彦 (課題代表者) : 地震・津波の予測精度の高度化に関する研究 (hp120312) , 平成 24 年度 HPCI システム利用研究課題利用報告書 (2013).

(2014 年 2 月 28 日受付)

安藤和人 kazutoa@jamstec.go.jp
(独) 海洋研究開発機構地球情報基盤センター所属. HPCI 戦略分野 3 における研究活動のサポート業務に従事.

馬場俊孝 babat@jamstec.go.jp
(独) 海洋研究開発機構地震津波海域観測研究開発センター所属. 博士 (理学). 専門は津波, 津波防災.

松岡大祐 daisuke@jamstec.go.jp
(独) 海洋研究開発機構地球情報基盤センター所属. 博士 (工学). 大規模データ可視化に関する研究開発に従事.

加藤季広 t-katou@ew.jp.nec.com
日本電気 (株) 第一官公ソリューション事業部所属. 科学技術計算アプリケーションの性能最適化, 開発支援業務に従事.
