

デスクトップ PC における実用的な津波数値計算の高速化手法

岩手大学 正会員 ○岩間 俊二
岩手大学 正会員 小笠原敏記
岩手大学 正会員 柳川 竜一

1. はじめに

GPU コンピューティングやスーパーコンピュータによる大規模並列計算によって、津波数値計算のリアルタイム化が進んでいる。しかし、大規模なプログラムチューニングが施された津波解析コードは、ポータビリティ及びコストの面で一般向きとは言い難く、デスクトップ PC を用いた津波数値計算への転用に課題が残る。デスクトップ PC 向け CPU のマルチコア化は、2005 年のデュアルコア（Intel Pentium D）から始まり現在 8 コア（同 Core i7-5960X）まで拡大され、SIMD 拡張命令の進化（SSE, SSE2, SSE3, SSE4, AVX, AVX2）と合わせて高速な並列計算実行環境が整いつつある。本研究では、津波解析コードの改変を最小限に抑えつつ計算実行速度向上を図ることを目標に、インテル社 C/C++コンパイラによるコード最適化手法を検証し、デスクトップ PC における実用的な津波数値計算高速化の可能性について論ずる。

2. 津波数値計算の概要

津波数値計算は、非線形長波式（浅水理論）を Staggered Leap-Frog 法によって差分化したプログラムを用いた。計算領域は、波源域 1080m 正方メッシュから遡上域 10m 正方メッシュ（岩手県山田町）までネスティングさせた地形を用いし、東北地方太平洋沖地震津波を再現計算した（図 1、表 1）。

表 1 計算領域諸元

	$\Delta x(m)$	横×縦格子数	$\Delta t(sec)$
A 領域	1080	613×953	1.00
B 領域	360	492×1032	1.00
C 領域	120	306×306	1.00
D 領域	40	552×666	0.50
E 領域	10	1148×1100	0.10
波源モデル	東北大学モデル version1.1		
地形モデル	平成 16 年度岩手県総合防災室データ		

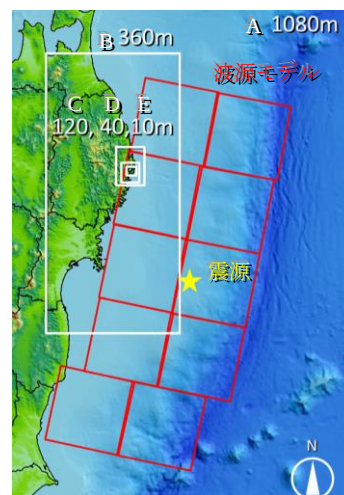


図 1 計算領域配置（東北地方）

3. 数値計算実行環境及び高速化手法

数値計算実行環境は一般的な 64bit Windows デスクトップ PC とし、表 2 の通りである。C/C++ 言語で記述した津波解析コードは、最適化なし、最適化あり、自動ベクトル化、自動並列化、OpenMP 並列化に分けてそれぞれコンパイルし、計算結果に有意な差が出ないことを確認したうえで、再現計算 3600 秒にかかる実行時間を計測した。OpenMP は、津波解析コード内の計算負荷の高いループ文手前に OpenMP 指示文を 1 行挿入するだけの簡単なものである。他の最適化オプションに切り替えた場合、OpenMP 指示文は、コメント行とみなされるため、同一コードによる一貫した実行速度の検証が行える。また、単精度と倍精度浮動小数点演算の実行速度の違いも比較検証した。

表 2 数値計算実行環境

OS	Microsoft Windows 8.1 64bit
Compiler	Intel C/C++ ComposerXE2013
OpenMP	Version 3.1
CPU	Intel Core i5-4570(3.2GHz 4Core)
MEM	8Gbytes
HDD	1TBytes

4. 高速化手法の違いによる実行速度の検証

3600 秒の再現計算に要した実行時間は表 3 の通りである。OpenMP 並列化は、最適化のみ（ベクトル化なし）に対して約 2.66 倍の速度向上となった。4 コア CPU では理論上 4 倍の速度向上が期待されるが、並列スレッド生成に関わるオーバーヘッドは比較的重い処理とされ、並列計算量が少ない場合、並列計算効果よりも並列スレッド生成に要する時間が相対的に増え、4 倍未満の速度向上率となるのが普通である。次に最適化のみに対する OpenMP 並列化実行速度向上率を計算領域別及び計算ルーチン別に示したものが表 4 である。計算時間刻み (Δt) が小さく、総格子数の多い E 領域が計算時間の大半を占めており、同領域の地殻変動計算ルーチンは OpenMP 並列化で 3.76 倍に高速化されている。流量計算ルーチンは 3.02 倍と低下するが、これは同領域が極浅海域における移流項と底面摩擦項及び遡上境界条件処理などをループ文中で複雑に判定処理するため CPU キャッシュミスが増え、並列計算効率を落としているものと考えられる。

一方、沖合 A 領域の流量計算ルーチンは遡上計算が無いため、3.92 倍と理想的な並列計算効率になっている。

自動ベクトル化及び自動並列化は、最適化のみの場合とほぼ同じ実行時間であった（表 3）。倍精度と単精度の浮動小数点演算の実行速度も比較したが、両者ともほぼ同じ実行時間であったことから、自動ベクトル化が有効に機能していないことが裏付けられる。ベクトル化は 128 ビット SIMD 命令 1 つで倍精度浮動小数点演算を 2 つ同時に実行（単精度では 4 つ）できる CPU レジスタ内部の並列演算機構である。ベクトル化を阻害する要因を見つけて対処すれば、倍精度で理論上 2 倍（単精度で 4 倍）、256 ビット SIMD 命令を使えば更に 2 倍の高速化が見込める。現状では、SIMD 組み込み関数（SIMD Intrinsic）を利用することで高速化が可能であるが、コード改変に伴う見通しの悪さと移植性に課題が残る。このため、最新の OpenMP Version4.0 以降で採用された SIMD 構文を活用して、オリジナルコードの改変を最小限に抑えて高速化することが望ましい。

4. まとめ

津波解析コードの実行速度の違いを C/C++ コンパイラの最適化設定により比較検証した。OpenMP 並列化設定は、マルチコア CPU における並列計算を比較的簡単に実現できることを確認した。自動並列化及び自動ベクトル化は現状のコードでは有効に機能しないことが確認されたが、OpenMP Version4.0 以降の Parallel for 構文及び SIMD 構文を組み合わせれば、単精度浮動小数点演算の場合、8 コア CPU と 256 ビット SIMD 命令（1 命令で同時に 8 データ処理）によって、理論上 64 倍高速に計算できる可能性が出てきた。現実はいずれより並列効率は低いと考えられるが、ポータビリティ性及びコスト面において有望な高速計算手法と考えられる。

参考文献

- ・越村俊一，香月恒介，茂渡悠介，GPU コンピューティングによる津波解析の高速化とリアルタイム浸水予測，土木学会論文集 B2（海岸工学論文集，第 57 巻），Vol. 66, No.1, pp.191-195, 2010.
- ・後藤智明，小川由信：Leap-Frog 法を用いた津波の数値計算法，東北大学工学部土木工学科，52p，1982.
- ・北山洋幸：OpenMP 入門，秀和システム，2009.

表 3 最適化別の実行速度

最適化	実行時間（秒）	コンパイラ設定
なし	8701	Od
最適化のみ	2883	02 Qvec-
自動ベクトル化	2797	02 QxSSE4.2
自動並列化	2789	02 Qparallel
OpenMP 並列化	1085	02 Qopenmp

表 4 計算領域別及び計算ルーチン別の実行時間（上段：最適化のみ，下段：OpenMP 並列化）

	A領域	B領域	C領域	D領域	E領域
地殻変動計算 (Okada1992)	3.7 1.0	3.2 0.8	0.8 0.2	2.3 0.6	7.9 2.1
水位計算 (連続式)	11.1 5.4	7.5 5.2	1.7 0.8	10.4 6.3	178.2 120.4
領域接続 (水位)	3.5 1.3	0.8 0.2	1.9 0.9	10.0 6.1	
流量計算 (運動の式)	98.3 25.1	64.3 25.5	13.1 5.3	84.1 24.8	1660.0 550.3
領域接続 (流量)	5.2 2.1	1.7 0.9	3.2 1.4	16.8 8.6	4.1 3.5
最大値更新	6.3 3.8	4.7 3.0	1.0 0.7	6.2 3.7	643.4 252.8
スナップショット 出力他	1.9 2.0	0.0 0.0	0.0 0.0	0.0 0.0	22.9 17.9
既知量と未知量 の交換	0.0 0.0	0.0 0.0	0.0 0.0	0.0 0.0	0.0 0.0
合計	130.0 40.7	82.3 35.8	21.6 9.2	129.8 50.1	2516.6 947.1

(秒)