

高解像度 LES 計算の GPU による高速化と性能評価

*二星 義裕, 朴 泰祐, 埴 敏博 (筑波大学システム情報工学研究科)

池田 亮作, 日下 博幸 (筑波大学生命環境科学研究科)

1 はじめに

近年, GPU (Graphics Processing Unit) の持つ高い浮動小数点演算処理能力と高いメモリバンド幅が注目され, GPU をグラフィックス計算以外の汎用計算に用いる GPGPU (General-Purpose GPU) の研究が盛んに行われている. 汎用 CPU と比べ GPU は非常に高い並列性・演算性能・メモリ転送性能を備えていることから, 数値流体力学をはじめ, 分子動力学, 重力多体計算や高速フーリエ変換など GPU を利用した研究が精力的に進められている [1, 2].

気象計算の分野においても, 高速な計算処理が要求されている. 東京工業大学学術国際情報センターでは, 工学系の数値流体力学モデルとしての LES モデル (Large Eddy Simulation) [3] や気象庁が開発を進める次世代気象計算のプロダクション・コード ASUCA の力学過程を含むコードを GPU に対応させている [4]. 一方, 空間を詳細に分解した都市気象モデルの GPU 化は, ほとんど例がない. そこで, 本研究では, 空間分解能 10m の LES モデルをベースにした都市気象モデルを GPU 計算に対応させる.

2 背景

近年, 都市のヒートアイランド研究などにおける気象モデルの高解像度計算の一手段として LES が注目されている. LES 計算は, 理想計算を対象としてきたため地形の導入はなく地面は平坦であるものが多い. そこで筑波大学の池田らによって, 複雑な地形の効果を取り入れた一般曲線座標系による LES 気象モデルが開発されてきた [5, 6]. このような気象モデルでは計算量が多く, 計算結果を得るまでに長い時間を必要とする. 一方, 下川辺らによる研究では気象計算の GPU 化を行っており, 気象分野における GPU 処理の有効性が示されている. そこで, 本研究では池田らの開発した気象モデルを GPU に対応させ計算時間を短縮する.

3 LES 気象モデル

池田らの開発した気象モデルは, 地形の効果を取り入れている. 地形を表現できる LES として, 直交座標系を採用した Raasch and Schroter (2001) のモデルや, 地形に沿った座標系を用いた Chow et al. (2006) のモデルなどがある. 後者の座標系の場合, 急峻な地形に対しては座標変換誤差が大きくなることが指摘されていることから, 池田らの開発した LES モデルには一般

曲線座標系を導入している. このモデルは, 筑波大学におけるスーパーコンピュータ T2K-Tsukuba 上で開発されてきた. 本研究では, 池田らの開発した都市・複雑地形を対象とした LES 気象モデルを GPU に対応させる. 研究で扱う LES の数値計算アルゴリズムは SMAC 法で, 移流項に 2 次精度 Adams-Bashforth 法, 拡散項に Crank-Nicolson 法を用いている. ポアソン方程式は Bi-CGStab 法で解いている.

4 GPU

GPU は, 本来画像処理のための補助演算装置である. CPU に比べ高い演算性能を持つようになったため, GPU の演算資源を画像処理以外の目的に応用する技術である GPGPU が数値シミュレーションなど幅広い分野で利用されている. 代表的な GPU である NVIDIA 社の CUDA アーキテクチャでは, SM (Streaming Multiprocessor) と呼ばれるマルチプロセッサが複数並んだ構成をとっている. この場合, 一つの SM には SP (Streaming Processor) と呼ばれるコアが 8 個とシェアード・メモリと呼ばれるデータ共有のための高速なオンチップメモリを持っている. 一方, GPU の全体メモリとして大容量のグローバルメモリがあるが, チップ外のメモリのため, データアクセスはシェアード・メモリに比べ低速である. 本研究で使用する新世代 CUDA アーキテクチャ “Fermi” では, グローバルメモリのキャッシュが搭載されアクセス性能が向上している.

5 GPU による高速化の検討

LES における流体計算は基本的にはいわゆるステンスル計算であり, 領域分割法における内点計算と境界データの交換のコスト比率の観点から, GPU のような加速演算装置を使うメリットがあり, CPU に比べ大幅な高速化が期待できる. 本研究で扱う気象モデルの高速化を行うため, まずコード内の各サブルーチンが占める実行時間のプロファイリングを行った. プロファイリング結果を図 1 に示す. プロファイリングの結果に基づき cgstab (Bi-CGStab 法でポアソン方程式を解くサブルーチン), addition_inst_value (時間平均量を求めるため, 瞬時値の加算するサブルーチン) 以外のサブルーチンについて CPU 実行の場合の全時間の 70.58% が消費されていることがわかった. このプロファイリングの結果に基づき, cgstab, addition_inst_value 以外のサブルーチン (移流項, 拡散項を計算するため

のサブルーチン)と一部の処理について GPU を用いるために CUDA コードの実装を行った [7] .

Each sample counts as 0.01 seconds.						
%	cumulative	self	self	total		
time	seconds	seconds	calls	Ks/call	name	
25.80	35022.13	35022.13	38233	0.00	0.00	__module_bicgstab_MOD_cgstab
24.56	68357.84	33335.71	191165	0.00	0.00	__module_dynamics_MOD_gradient_cell_center_surface
16.44	90682.76	22324.92	1	22.32	135.76	__module_run_MOD_run
11.55	106368.40	15685.64	76466	0.00	0.00	__module_dynamics_MOD_gradient_cell_surface
6.62	115356.29	8987.89	38233	0.00	0.00	__module_sg_s_MOD_sg_s_stress_vec
2.98	119395.75	4039.46	38233	0.00	0.00	__module_smac_MOD_smac
2.41	122667.01	3271.26	20000	0.00	0.00	__module_addition_inst_value_MOD_addition_inst_value
2.23	125691.93	3024.93	38233	0.00	0.00	__module_sg_s_MOD_sg_s_stress_sca
2.00	128406.13	2714.19	38233	0.00	0.00	__module_dynamics_MOD_tke_flux
1.34	130228.95	1822.82	191165	0.00	0.00	__module_dynamics_MOD_diffusion_crank_nicolson
0.86	131390.48	1161.53	38233	0.00	0.00	__module_dynamics_MOD_gradient_pres
0.84	132535.98	1145.50	100000	0.00	0.00	__module_dynamics_MOD_advection_adams_bashforth_2nd
0.81	133630.44	1094.46	20000	0.00	0.00	__module_dynamics_MOD_contravariant_velocity
0.35	134103.40	472.96	38233	0.00	0.00	__module_dynamics_MOD_gradient_scale

図 1: 都市気象モデル LES 計算のプロファイリング
(文中では module....MOD_は省略する)

表 1: 評価環境

CPU	Intel Xeon E5630 2.53GHz 4coresx2
RAM	DDR3 SDRAM 1066MHz 4GBx6 GDDR5 SDRAM 1.55GHz 3GB (ECC on)
GPU	NVIDIA Tesla M2050 1.15GHz
OS	CentOS Linux release 6.0 (Final)
Compiler	nvcc 4.0 (-arch sm_20) for GPU code

6 性能評価

プロファイリング結果にある cgstab, addition_inst_value 以外のサブルーチンと run の一部の処理に関して GPU 対応し, その処理の部分に関して CPU の単一コアで行った場合と GPUで行った場合の性能比較を図 2 に示す. 評価環境を表 1 に示す. Tesla M2050 では, 共有メモリ 16KB / L1 キャッシュ 48KB, または共有メモリ 48KB / L1 キャッシュ 16KB の構成が可能である. 本研究では, 前者の構成で性能評価を行った. 縦軸は実行時間. 横軸は問題サイズを表す. 問題サイズ $N = x_{max} \times y_{max} \times z_{max}$ とし, ここでは, $z_{max} = 102$ と固定し x_{max} と y_{max} のサイズを変化させた場合の実行時間の変化を示している. なお, GPU のメモリが 3GB であるため, GPU 上で実行できる x_{max} , y_{max} の問題サイズは 132 までに制限される. 図 2 より, 全ての問題サイズにおいて GPU の速度が CPU を大幅に上回ることが確認できた. 問題サイズ (x_{max} , y_{max}) が 102 の場合と 132 の場合で, それぞれ 7.9 倍, 8.4 倍の速度向上が達成された. これは, GPU の倍精度演算性能が大幅に向上したこと, また本研究で扱った LES モデルはデータ参照が多いステンシル計算であるため, GPU の高いメモリバンド幅が有効であり, CPU に比べ処理時間を大幅に短くすることができたためと考えられる.

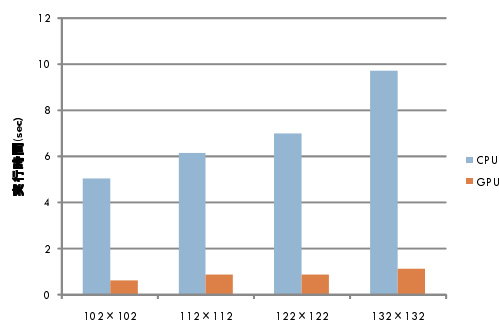


図 2: X, Y 方向における問題サイズに対する処理時間

7 まとめと今後の課題

都市気象モデルを対象とした LES 計算において, 計算負荷の高い関数を GPU に対応させ, 計算処理時間の短縮を実現した. 今回は, 比較的小さな問題サイズであったため, 大規模な問題サイズを扱えるように, GPU を搭載した複数ノードを用いる並列化を行い, 性能評価を行う予定である.

参考文献

- [1] 額田 彰: CUDA による高速フーリエ変換, Vol 20, No.2, pp. 37-43. 応用数学会. Jun. 2010.
- [2] 濱田 剛, 似鳥 啓吾, 青木 尊之: TSUBAME GPU クラスタを用いた重力多体シミュレーションの性能評価, 計算工学講演会論文集 (日本計算工学会). May. 2009.
- [3] 小野寺 直幸, 青木 尊之, 小林 宏充: GPU によるラージエディ・シミュレーションの高速化, 流体力学会年会 2010, 日本流体力学会. Dec. 2010.
- [4] 下川辺 隆史, 青木 尊之, 石田 純一, 河野 耕平, 室井 ちあし: メソスケール気象モデル ASUCA の TSUBAME2.0 での実行, 日本流体力学会 第 24 回数値流体シンポジウム講演予稿集. Dec. 2010.
- [5] 池田 亮作, 日下 博幸, 飯塚 悟, 朴 泰祐: 一般曲線座標系による並列 LES モデルの開発, 日本気象学会 2011 年度春季大会講演予稿集. May. 2011.
- [6] Ryosaku Ikeda, Hiroyuki Kusaka, Satoru Iizuka, Taisuke Boku: Development of Local Meteorological Model based on CFD, 5th International symposium on wind effects on buildings and urban environment (ISWE5). Mar. 2011.
- [7] NVIDIA Corporation: CUDA ZONE http://www.nvidia.com/object/cuda_home.html.