1G-03

データ同化処理における時空間ブロッキングの異機種性能評価

藤川 隼人† 池田 朋哉‡ 片桐 孝洋†3 永井 亨†3 荻野 正雄†3

名古屋大学工学部電気電子・情報工学科†

名古屋大学大学院情報科学研究科情報システム学専攻‡
名古屋大学 情報基盤センター†3

データ同化処理で用いられる計算方法の 1 つとして，計算コストが比較的少なくて済む Adjoint 法がある．我々は，Adjoint 法の中の 1st-order-adjoint と呼ばれる方式に対し時空間ブロッキングを適用することで，より性能のよいアルゴリズムの開発を試みている．しかし現在，限られたマシン上でしか性能評価が行われていないため十分ではない．そこで本論文では，この時空間ブロッキングを様々なマシンで動作させ，時空間ブロッキングで必要となる多種の性能パラメタを変化させることで達成されるチューニングの効果を検証する．また，計算機の特徴と時空間ブロッキング上現れる性能パラメタを用いた実行時間のモデル化の検討も行う．

1. はじめに

大規模数値シミュレーションと大容量観測データの融合を図る計算技術として，データ同化が注目されている．データ同化は，現代の気象予報，海洋学などで活用されている．

データ同化の手法として Adjoint 法がある．この Adjoint 法の問題点として，メモリアクセスの増大による演算効率低下がある． Adjoint 法の Forward model の計算では差分法を用いるが，差分法は近傍の格子点値を使用するステンシル計算を用いた計算方法であるため，大規模で自由度が高いモデルでは，近傍の格子点値のアクセス範囲がキャッシュ容量を超えてしまう．その結果，メモリアクセスが増加し，演算効率低下につながる．これを防ぐためには，高速メモリであるキャッシュ上のデータを出来るだけ再利用して演算を行うブロッキングが必要となる．

ブロッキングによる高性能化について，Adjoint 法の Forward model におけるステンシル計算に時空間ブロッキングを適用する手法が提案されている[1][2]．しかし，この手法の性能評価は限られたマシン上でしか行われておらず性能評価が十分とはいえない．そこで本論文では，時空間ブロッキングを様々なマシン上で動作させ，性能パラメタを変化させることでチューニングの効果を検証した．

2. 時空間ブロッキング

Adjoint 法は，Forward model および Backward model を利用して推定を行う手法である．本論文では Forward model のみに対して評価を行う．ブロッキング無しの Forward model では，データ全体の計算を行った後に次のタイムステップへ進むが，大規模データにおいてキャッシュ上のデータの再利用ができない．そこで，キャッシュ上のデータが再利用しつつ，先に次のステップの計算を可能とするのが時空間ブロッキングの考え方である．ここで，時空間ブロッキングのサイズ（ブロックサイズ）とは，時間ステップにおいて何ステップ先まで計算を行うかの数値である．

3. 性能評価

名古屋大学情報基盤センターの Fujitsu PRIMEHPC FX100 及び Fujitsu PRIMERGY CX400 を用いた．両マシンとも計算ノードは NUMA (Non-Uniform Memory Access)を採用した 2 ソケットの構造とみなせるため，これを考慮して実験を行う．問題設定は FX100，CX400 ともに表 1 の条件で行った．

表 1. 問題設定

格子点数	1600 × 1600
初期値	すべて 0.2
真の値	乱数[0,1]の 0.1 刻み

実験は，FX100，CX400 ともにスレッド数とブロックサイズを変化させて行った．スレッド数は，FX100 では 1,4,16,25 と変化させ，CX400 では 1,4,14,16,25,28 と変化させた．CX400 でスレッド数 14 と 28 を追加したのは，1 ソケット 14 コアの NUMA の影響を考慮したからである．時空間ブロックサイズはスレッド数によって取れる範囲が違うため，スレッド数によって変化させるものとする．(ただし最大は 50 とした)．

本論文では，3 つの結果について考察する．はじめに CX400 においてブロッキングサイズを変化させたときの速度向上率，次に CX400 のスレッド数変化における台数効果，最後に CX400 と FX100 における実行時間の比較である．

Performance Evaluation of Space-time Blocking for Data Assimilation Processing with Various Machines

†Hayato Fujikawa, Electrical and Electronic Engineering and Information Engineering, School of Engineering, Nagoya University

‡Tomoya Ikeda, Graduate School of Information Science, Nagoya University

†3 Takahiro Katagiri, Toru Nagai, Masao Ogino, Information Technology Center, Nagoya University

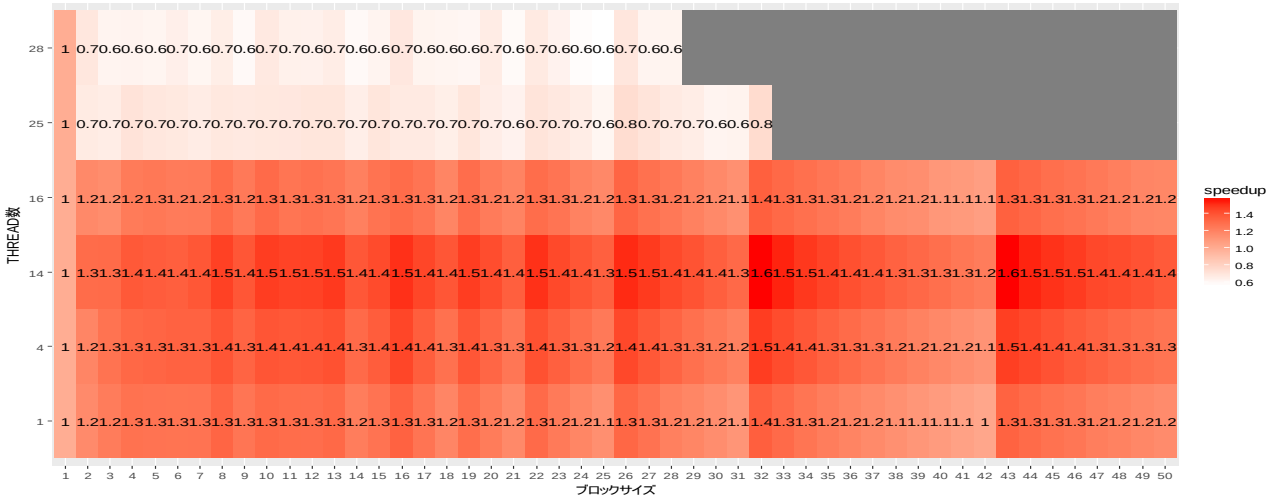

図 1. CX400 ヒートマップ

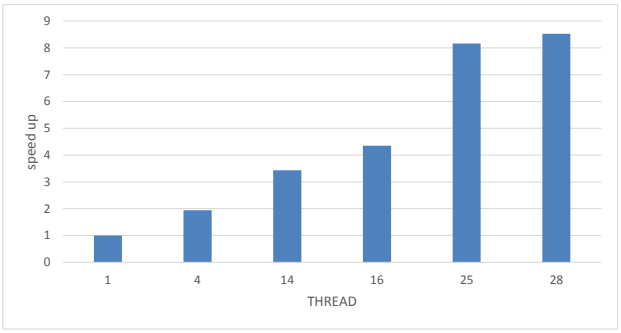

図 2. CX400 の台数効果

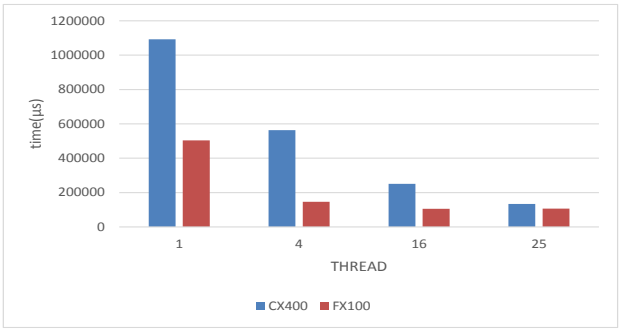

図 3. FX100 と CX400 の実行時間比較

まず，ブロックサイズを変化させたときの速度向上率について述べる．この結果は，図 1 にヒートマップとして示した．図 1 は，各スレッドにおいてブロックサイズが 1 の時（時空間ブロッキングを行っていない時）を基準とした速度向上率である．もっとも顕著な違いは，スレッド数「1, 4, 14, 16」と「25, 28」の間で起きている．前者はブロッキングを施した場合スピードアップしているが，後者は逆にスピードが遅くなっている．原因として，CX400 ではスレッド数 14 を境にソケットを跨ぐため，後者では NUMA を生じることが考えられる．

次に CX400 のスレッド数変化による台数効果について述べる．この結果を図 2 に示し，速度向上率のグラフとして記載した．各スレッドで一番速度が速いブロックサイズを用いている．スレッド数が増加すればするほどスピードアップが達成されているため，台数効果の観点でスケーラブルな結果を得た．

最後に，CX400 と FX100 の実行時間の比較について述べる．この結果を図 3 に示し，実行時間のグラフとして記載した．図 3 は図 2 の結果同様，各スレッドにおいて一番速度が速いブロックサイズを使用している．図 3 から，スレッド数「1, 4, 16」の時は FX100 のほうが CX400 よりもそれぞれ 2.17 倍，3.86 倍，2.37 倍速いが，スレッド数「25」の時は 1.26 倍 FX100 が速い．FX100 と CX400 の 1 ノードでの計算性能を FLOPS で比較したとき，理論的には CX400 が FX100 より約 1.16 倍高性能であるので，この結果については具体的な検証が必要である．

4. まとめ

今回行った検証では，CX400 における時空間ブロッキングの効果検証，および FX100 と CX400 での性能差について確認することができた．現段階で CX400 ではスレッド数が多くなると時空間ブロッキングの効果が出ない．また，少ないスレッド数の時には CX400 の速度は FX100 のほうが約 2.2 倍～3.9 倍速い．これらの性能差が生じる原因は解明されていないため，今後も性能評価を重ねて検証を続けていくことが必要である．

今回の性能評価では時空間ブロッキングのみを適用したモデルで評価を行っている．しかし，複数の Forward model の計算をブロッキングして同時に計算する 2 段の階層ブロッキング手法[1]が存在するため，この手法についても性能評価を行う必要がある．また，最終的に性能モデル化を行うことが，今後の重要な課題である．

謝辞

本研究の一部は，科学技術研究費補助金基盤研究（B）「通信回避・削減アルゴリズムのための自動チューニング技術の新展開」（課題番号：16H02823）による．

参考文献

[1] 池田朋哉 ほか，アジョイント法における Forward model への階層ブロッキング適用による高性能化，情報処理学会研究報告，Vol. 2016-HPC-157, No. 17, pp. 1-8 (2016)
[2] K. Datta, et al., "Stencil computation optimization and auto-tuning on state-of-the-art multicore architectures," Proceedings of the 2008 ACM/IEEE conference on Supercomputing, IEEE Press, 2008.