実時間シミュレーションへの応用を前提とした SMW 公式を用いた逆行列計算のハイブリッド並列による高速化

岩永 翔太郎・福間 慎治・森 眞一郎
(福井大学大学院工学研究科 情報・メディア工学専攻)

1 研究の背景と目的

本研究では, 多くのシミュレーション中に現れる線型方程式 $\boldsymbol{Ax} = \boldsymbol{b}$ の求解問題に着目する. 特に, シミュレーション中のユーザからのインタラクションや時間経過によって係数行列 $\boldsymbol{A}$ の一部の要素のみが変化をする状況でのリアルタイムシミュレーションの実現を研究のターゲットとする.

2 研究の概要

一般に, 連立一次方程式の求解問題では係数行列 $\boldsymbol{A}$ の逆行列を直接求めて解を導出することは稀である [2]. しかしながら,$\boldsymbol{A}$ の近似行列の逆行列が既知である場合, この逆行列を用いた補正計算により高速に $\boldsymbol{A^{-1}}$ を求める方法が存在すれば $\boldsymbol{A}$ の逆行列を導出して方程式の解を求める方法も十分に有効な手法となる.

このような目的で使用可能な数学公式の 1 つに, SMW 公式 (Sherman-Morrison-Woodbury foumula)[1] がある. この SMW 公式は次のような式で表される.

$$(\boldsymbol{A}+\boldsymbol{BC})^{-1} = \boldsymbol{A^{-1}} - \boldsymbol{A^{-1}B}(\boldsymbol{I}+\boldsymbol{CA^{-1}B})^{-1}\boldsymbol{CA^{-1}} \quad (1)$$

ここで $\boldsymbol{A}$ は $n \times n$, $\boldsymbol{B}$ は $n \times s$, $\boldsymbol{C}$ は $s \times n$ の大きさの行列で, $\boldsymbol{A}$ は正則であるとする. すなわち,$\boldsymbol{A}$ の近似行列 $\boldsymbol{A'}$ が $\boldsymbol{A^{-1}}$ を用いて補正により求まる可能性を示唆している. $\boldsymbol{A'} = \boldsymbol{A} + \boldsymbol{\Delta A}$ と表現し, さらに $\boldsymbol{\Delta A}$ を $\boldsymbol{\Delta A} = \boldsymbol{\Delta A_c} \times \boldsymbol{I} \times \boldsymbol{E_c}$ とする. ここで,$\boldsymbol{B} = \boldsymbol{\Delta A_c}$, $\boldsymbol{C} = \boldsymbol{E_c}$ と考えると, まさしく式 (1) より $\boldsymbol{(A')^{-1}}$ が求まることになる (式 (2)).

$$(\boldsymbol{A}+\boldsymbol{\Delta A})^{-1} = \boldsymbol{A^{-1}} - \boldsymbol{A^{-1}\Delta A_c}(\boldsymbol{I}+\boldsymbol{E_cA^{-1}\Delta A_c})^{-1}\boldsymbol{E_cA^{-1}} \quad (2)$$

式 (2) において, 右辺第 2 項の行列計算のサイズは図 1 のようになる. $\boldsymbol{(I + A_cA^{-1}\Delta A_c)^{-1}}$ の計算に $O(s^3)$ の時間がかかるが,$n >> s$ かつ s を定数とみなせる場合, 全体としては $O(n^2)$ となる.

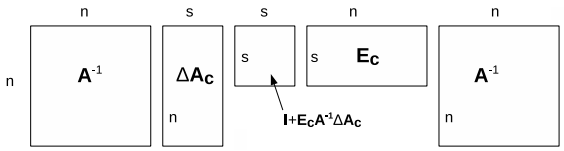

図 1: SMW 公式の行列積部分の行列サイズ

SMW 公式での計算行程では行列積計算が多くの割合を占め, 並列処理による高速化が得られる. また,$\boldsymbol{\Delta A_c}$ と $\boldsymbol{E_c}$ は疎行列であるため,$\boldsymbol{A}$ が疎行列か密行列かによらず疎行列性が利用できる. 我々の研究では, スレッド並列による並列処理と疎行列圧縮を併用し, 以下の手順で計算を行うことで,4.6 倍の高速化を達成した [3].

1. $\boldsymbol{S_1} = \boldsymbol{A^{-1}} \times \boldsymbol{\Delta A_c}$
2. $\boldsymbol{S_2} = \boldsymbol{E_c} \times \boldsymbol{S_1}$
3. $\boldsymbol{S_3} = \boldsymbol{I} + \boldsymbol{S_2}$
4. $\boldsymbol{S_4} = \boldsymbol{(S_3)^{-1}}$ を直接法で計算
5. $\boldsymbol{S_5} = \boldsymbol{E_c} \times \boldsymbol{A^{-1}}$
6. $\boldsymbol{S_6} = \boldsymbol{S_4} \times \boldsymbol{S_5}$
7. $\boldsymbol{S_7} = \boldsymbol{S_1} \times \boldsymbol{S_6}$
8. $\boldsymbol{S_8} = \boldsymbol{A^{-1}} - \boldsymbol{S_7}$
9. $\boldsymbol{S_9} = \boldsymbol{S_8} \times \boldsymbol{b}$

しかし, ステップ 7,8 について十分な並列化効果が得られず, 実行時間全体の 95%を占めていることを確認した. これは計算時に行列が疎行列圧縮できず, メモリアクセスがボトルネックになっていると考えられる. そこで本研究では, 複数の計算ノードを用いて並列化し, 利用できるメモリバンド幅を増やすことによる高速化効果について検討する.

3 ハイブリッド並列による高速化

複数の計算ノードを用いて並列化することで, 利用できるメモリバンド幅を増やし高速化を図る. ノード間での通信量を抑えるため, ステップ 1〜6 は各ノードが全領域を計算し, ステップ 7〜9 を MPI を用いたノード間並列で計算を行った. この方法では, 全ノードに $\boldsymbol{b}$ を送信ならびに各ノードから $\boldsymbol{x}$ を受信するタイミングでしか通信は発生しない. 更に各ノード内で,OpenMP を用いてステップ 4 以外を並列化する. 実験の実行環境は CPU:Core2Duo 2.66[GHz],Memory:2[GB],L2 Cache:4[MB], OS:Linux 2.6.19-1.2895.fc6 を 8 ノード用意し,Gigabit Ethernet で結合した. コンパイラは icc 12.0 (-O3 オプション + MKL10.0.3.020) を用いた. 係数行列 $\boldsymbol{A}$ のサイズ n は 3759, 非零要素の割合は 1.00%の行列である. サイズ s については 32 で固定した. ステップ 1〜9 までを合計した実行時間を図 2 に示す.

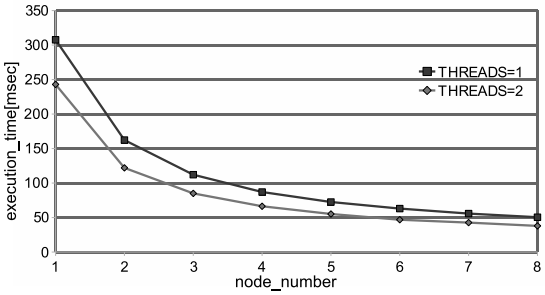

図 2: 複数ノードで並列計算時の実行時間

ノード間並列計算を用いることで, 計算時間の 96.9%を占める部分を並列処理することができ, ノード数が 1 の場合と 8 の場合を比較すると 6.1 倍の高速化が得られた. スレッド並列の効果としては 1.3 倍程度の高速化が得られ, ハイブリッド並列を行うことで全体として 8.1 倍の高速化を達成した.

4 考察

SMW 公式の計算において,1 台のノード内でのマルチコア並列処理では十分な高速化が得られなかったステップ 7,8 に対して複数の計算ノードを用いた並列計算を行うことで, それぞれ 8.0 倍,7.8 倍の高速化が得られた. このことから, メモリアクセスの競合が十分な高速化が得られなかった原因であることが確認できた. ノード間の通信時間も 8 ノードの場合で 3[msec] 程度と, 得られた高速化効果に対して微々たるものであると考えられる. ハイブリッド並列を行うことで実行時間全体を 8 倍以上高速化することができ, ハイブリッド並列が有効な手法であることを確認した.

謝辞

本研究の一部は, 科学研究費補助金 (基盤研究 (S)16100001, 基盤研究 (C)22500044) の補助による.

参考文献

[1] G.H. Golub and C.F.Van Loan, Matrix Computations, 3rd ed., Johns Hopkins Univ. Press, 1996.

[2] 二宮市三, 吉田年雄, 長谷川武光, 秦野やすよ, 杉浦 洋, 櫻井鉄也：数値計算のつぼ (2004)

[3] 岩永翔太郎, 福間慎治, 森眞一郎, “実時間シミュレーションへの応用を前提とした SMW 公式を用いた逆行列計算のマルチコア並列処理” 電子情報通信学会論文誌 2011/7 Vol.J94-D No.7 pp.1165-1168