

# 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処理の予備評価

松井 祐太・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

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  $Ax = b$  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  $A$  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。通常、連立一次方程式の求解問題では係数行列  $A$  の逆行列を直接求めて解を導出することは少ない。しかしながら、 $A$  の近似行列の逆行列が既知の場合、これを用いて高速に  $A^{-1}$  を求める方法が存在すれば  $A$  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) がある。 $A$  を  $n \times n$ ,  $\Delta A_c$  を  $n \times s$ ,  $E_c$  を  $s \times n$  の行列で、 $A$  は正則とすると  $A$  の近似行列  $A'$  を  $A' = A + \Delta A$  と表現し、さらに  $\Delta A = \Delta A_c \times E_c$  とすると SMW 公式より

$$(A + \Delta A)^{-1} = A^{-1} - A^{-1} \Delta A_c (I + E_c A^{-1} \Delta A_c)^{-1} E_c A^{-1}. \quad (1)$$

この式は、 $A^{-1}$  を用いて  $(A^{-1})'$  が求まる可能性を示唆している。 $(I + A_c A^{-1} \Delta A_c)^{-1}$  の計算に  $O(s^3)$  の時間がかかるが、 $n \gg s$  かつ  $s$  を定数とみなせる場合、 $O(n^2)$  となる。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。我々の研究では、マルチコアプロセッサを用いたノード内スレッド並列による並列処理と疎行列圧縮を併用し、次の手順で計算し、4.6 倍の高速化を達成した [1]。これを従来法とする。

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

しかし、ステップ 7, 8 についてスレッド並列による十分な並列化効果が得られず、実行時間全体の 95% を占めていることがわかった。この要因を調査した結果メモリボトルネックであることが判明した。

## 2 ハイブリッド並列処理

前述のボトルネックを解消するために、スレッド間並列とノード間並列を併用するハイブリッド並列処理を考える。手法として、メモリボトルネックを解消することを主眼とした Hybrid1 [2] と、計算全体を並列処理する Hybrid2 を考える。ここでは、Hybrid2 について述べる。いま、ステップ 1 からステップ 9 までを 1 ステージと定義する。Hybrid1 では、この 1 ステージを高速に求解することは可能である。しかし、計算開始時点で各ノードが、 $A^{-1}$  の全体を保持する必要がある。そのため、2 ステージ目に移るとき、各ノードで求めた  $A^{-1}$  の部分行列の gather が全ノードで必要 (Alltoall の通信) となる。このとき、1 ノードあたりの通信量は送受信合わせると  $O(n^2)$  となる。この通信量は非常に大きく、並列化に伴うメリットを失ってしまう。Hybrid2 では、このステージ間でのデータ再配置を不要とするものである。計算開始時点で、各ノードは  $A^{-1}$  を行方向ブロック分割した部分行列を所持している。Hybrid1 と同様のアルゴリズムを用いることで、次のステージで必要となる  $A^{-1}$  の部分行列が各ノードに存在することとなり、ステージ間でのデータ再配置が不要となる。一方で、 $A^{-1}$  がブロック分割して分散配置されることに対応して、ステップ 1~6 も  $A^{-1}$  の配置形態に対応したデータ並列処理の検討が必要となる。したがって、 $E_c$  は列方向のブロック分割を行い各ノードに配置するものとする。この際、ステップ 1~6 を全てデータ並列処理するとステップ間でのデータ再配置が必要となりボトルネックの要因となる。そこで我々はステップ 1, 2 および 5 のみをデータ並列処理、ステップ 3, 4 および 6 を重複処理させる方法を採用し、さらにステップ 2 および 5 の処理を、各ノードで部分和 ( $S_2'$  および  $S_5'$ ) を計算する処理と、デー

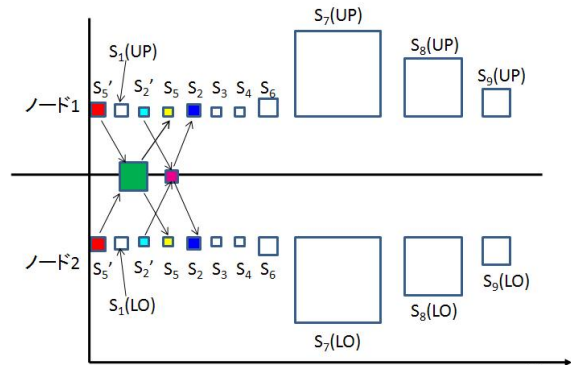


図 1: 提案法のタイムチャート

タ再配置を行う処理、部分和を合成する処理に分割し、データ再配置とその他の処理をオーバーラップ実行することで通信時間を隠蔽するための計算順序の入れ替えを行う手法を提案する。提案法のタイムチャートを図 1 に示す。

実験の環境は、CPU:Core2Duo(2.66[GHz],L2 Cache:4[MB]), Memory:2[GB], OS:Linux2.6.191.2895.fc6, ノード内並列度 2, コンパイラは icc12.0(-O3 オプション+MKL10.0.3.020) を 2 ノード用意し、Gigabit Ethernet で結合したものを環境 1 とし、CPU:Xeon X7560 (2.26[GHz],L3 Cache:24[MB])\*4, Memory:64[GB], OS:Linux2.6.18-23.1.1.el5 x86\_64, ノード内並列度 16, コンパイラは icc12.1(-O3 オプション+MKL10.3 Update6) を 2 ノード用意し、10Gigabit Ethernet で結合した。これを環境 2 とする。

行列  $A$  としては、問題サイズ  $n$  が 3759 の Liver (生体データ) と、23052 の bcsstk36 (MatrixMarket) を用いる。また非零要素の割合は liver が 1.0%, bcsstk36 が 0.2% である。サイズ  $s$  は 32 とする。実行時間を表 1 に示す。単位は全て msec である。

表 1: 実行時間

		逆行列計算		解計算		合計	
		環境 1	環境 2	環境 1	環境 2	環境 1	環境 2
Liver	従来法	250	27	24.4	2.6	274.4	29.6
	Hybrid1	106	16	12.8	1.4	118.8	17.4
	Hybrid2	105	15	13.1	1.4	118.1	16.4
bcsstk36	従来法		773		88		861
	Hybrid1		610		92		702
	Hybrid2		621		90		711

## 3 考察

Liver の場合、環境 1 において、ハイブリッド並列はスレッド間並列のみに比べて 2 倍以上の速度向上となった。一方、環境 2 においては利用できるメモリバンド幅およびメモリ速度向上のため約 1.8 倍の速度向上となっている。bcsstk36 も約 1.8 倍である。このことから、行列サイズが増えた場合でもハイブリッド並列は有効であることがわかる。Liver では Hybrid2 は Hybrid1 より若干速かったが、bcsstk では遅くなっている。Hybrid2 ではオーバーラップ実行により通信時間を隠蔽しているが、計算時間に比べて通信時間の伸び率の方が大きく、これが影響しているものと考えられる。しかしながら、Hybrid2 にはチューニングの余地が残っており、計算機にあわせたチューニングを行うことで通信時間を完全に隠蔽することが可能であると考えられる。したがって、今後本格的な評価を行う必要がある。

## 謝辞

本研究の一部は、JSPS 科研 基盤 (C)22500044 ならびに JST A-STEP (FS) 探索ステージ AS242Z00401H の助成による。

## 参考文献

- [1] 岩永ほか, “実時間シミュレーションへの応用を前提とした...” 電子情報通信学会論文誌 2011/7 Vol.J94-D No.7 pp.1165-1168
- [2] S. Iwanaga, S. Fukuma, S. Mori, “Hybrid Parallel Implementation of Inverse Matrix Computation by SMW Formula for Interactive Simulation” IEICE Trans. Info. & Sys., Vol.E95-D, No.12, pp2952-2953, Dec. 2012