スーパーコンピュータ FX10 を用いたマイクロマグネティックシミュレータの高速化

新井眞大，赤城文子，吉田和悦
(工学院大学大学院 工学科電気・電子専攻)

Speed-Up of STO Micromagnetic Simulator Using Supercomputer FX10

Masahiro Arai, Fumiko Akagi, Kazuetsu Yoshida
(Graduate School of Electrical Engineering and Electronics, Kogakuin University)

1．まえがき

近年，磁性材料を用いた様々な技術の設計手段には，磁性体の複雑な磁化挙動を解析するマイクロマグネティクスシミュレーションが用いられている．中でも，磁気ディスク装置(Hard Disk Drive: HDD)のヘッドや媒体それぞれの解析には非常に多く用いられている．一方，高記録密度化の手段として期待されているマイクロ波アシスト磁気記録方式(Microwave Assisted Magnetic Recording: MAMR) [1]は，磁気ヘッドと媒体を同時に解析することが重要である．しかし，要素数が 900 万個以上必要であり，高々4 ビット計算するにも通常のワークステーションでは数日を要する．

本研究では計算時間を短縮することを目的に，東京大学のスーパーコンピュータ FX10 を使用して並列化の手法を検討した．並列化手法として，メッセージ交換ライブラリに MPI(Message Passing Interface)を用いることで複数の CPU 間で並列処理を行う手法[2]，OpenMP を用いたマルチスレッドによる並列処理を行う手法[3]，そして上記二つの方法を合わせた Hybrid 並列化手法をシミュレータに導入して比較した．

2．計算モデル及び計算方法

高速化の検討のモデルは，MAMR のスピントルクオシレータ(STO)のみとした．STO は電流密度低減のためにスピン注入層を 2 層用いた構造とし，クロストラック方向と高さは 30 nm，STO 全体の膜厚は 25nm とした． 1 つの要素は 2.5 nm の立方体であり，要素数は 229376 個である．

本シミュレータでは，磁化挙動を(1)式に示す Landau-Lifshitz-Gilbert(LLG)方程式を解くことで求めた．

$$(1+\alpha^2)\frac{d\vec{M}}{dt} = -\gamma\vec{M}\times\left(\vec{H}_{eff}-\alpha\vec{H}_{st}\right)-\frac{\gamma}{M_s}\vec{M}\times\{\vec{M}\times\left(\alpha\vec{H}_{eff}+\vec{H}_{st}\right)\} \quad (1)$$

$\vec{M}$は磁化ベクトル，γはジャイロ磁気定数，αはダンピング定数，M_sは飽和磁化，$\vec{H}_{eff}$は実効磁界ベクトルとする．$\vec{H}_{st}$は STO へ電流を流す事で各層に印加される偏極スピンによる磁界(スピントルク磁界)である．また，実効磁界の一つである静磁界の計算に最も多くの計算時間を要するため，高速フーリエ変換(Fast Fourier Transform: FFT)を用いているが，計算時間の 70 %は静磁界の計算である．

Table.1　Specification of FX 10

Processor	1.848 GHz,16core ×1
Theoretical peak performance	236.5 Gigaflops
Memory capacity	32 GB
HDD/SSD	1.1 PB+2.1PB

計算に用いた東京大学のスーパーコンピュータ FX10 のスペックを Table.1 に示す．

3．プログラムの並列化

MPI の場合，反復処理の分割や通信命令をプログラミングしなくてはならない．まず，MPI 関数を用いて実行 CPU ノード数とランクを取得する．取得したノード数とランクを利用して処理範囲を各ノードに均等に割り当てるように分割することで反復処理の並列化を行う．その後，通信関数により各ノードの計算結果を統合することで，計算結果に矛盾が生じないようにする．通信関数には MPI_Allreduce()または MPI_Allgather()を用い，その性能を比較する．

OpenMP の場合は#pragma から始まる指示文を挿入するだけで並列化を行える．並列化を適用する for ループの直前に #omp parallel for schedule(static) private()を挿入することで処理範囲が均等に分割され，反復処理の並列化を行う．

Hybrid 並列化では MPI と OpenMP のそれぞれで並列化に必要な処理をプログラムに組み込み，並列化を行う．

また並列化は計算時間の大半を占める静磁界，磁化の FFT を行う計算部に対して行った．

4．計算結果

Fig.1 に OpenMP におけるスレッド数，MPI におけるプロセス数，及び hybrid 並列化におけるスレッド数（プロセス数は 7 に固定）と計算時間の関係を並列化の種類で比較した結果を示す．OpenMP は約 10.2 倍，MPI_Allreduce()は約 8.4 倍，MPI_Allgather()は約 11.8 倍，MPI_Allgather()を使用した hybrid 並列化は約 22 倍の高速化に成功した．このことから MPI と OpenMP の両方を利用した Hybrid 並列化が一番効果が得られることがわかった．また，MPI_Allreduce()では計算結果の統合の際に加算を行うため，その分通信時間がかかってしまったと考えられる．これより，演算を必要としないデータの統合を行う場合は MPI_Allgather()を用いた方が計算時間が短いことがわかった．

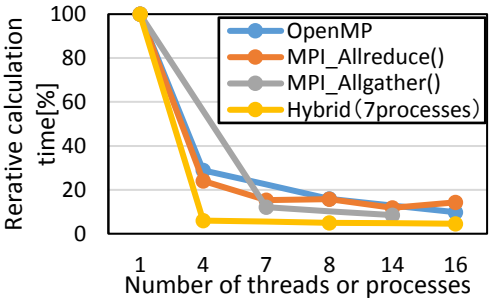

Fig.1 Relationship between number of threads or processes and calculation time for different types of parallelization

5．まとめ

本研究は東京大学のスーパーコンピュータ FX10 を用いてマイクロマグネティックシミュレータの並列化による高速化の検討を行った．静磁界の計算部に MPI_Allgather()を用いた Hybrid 並列化を適用することで計算時間を短縮できた．

謝辞　本研究の一部は，情報ストレージ研究推進機構(ASRC)の助成を受けて行った．

参考文献

[1] Y. Tang, and J. G Zhu, IEEE Trans. Magn. Vol. 44, no. 11, pp. 3376-3379, (2008) .
[2] P. Pacheco, 秋葉博: MPI 並列プログラミング, p. 43-56, 培風館, 東京, (2001)
[3] 牛島省: OpenMP による並列プログラミングと数値計算法, p. 11-73, 丸善, 東京, (2007)