

## ハイブリッド並列による Chebyshev 基底共役勾配法の性能評価

野村直也<sup>†</sup> 熊谷洋佑<sup>†</sup> 藤井昭宏<sup>†</sup> 田中輝雄<sup>†</sup>工学院大学<sup>†</sup>

## 1. はじめに

大規模な科学技術計算を行うにはスーパーコンピュータが用いられる。TOP500によると、近年のスーパーコンピュータの性能向上は、コア数増加が大きく寄与していることがわかる[1]。並列に処理を行う際にはノード間での通信が発生する。特にリダクションやブロードキャストなどの、全ノードがデータを送受信する集団通信は、ノード数が増えるほど通信時間が増えることが考えられる。

一方、正定値対称な疎行列を係数にもつ連立一次方程式を解くのに反復解法である共役勾配法 (Conjugate Gradient method : CG 法) [2]が広く用いられている。CG 法では1反復中に1回の疎行列ベクトル積と2回の内積計算が発生する。この内積計算では集団通信が発生し、特に高並列環境においてボトルネックとなることが考えられる。

そこで、Chebyshev 多項式を基底とし、Krylov 部分空間の生成をまとめることで集団通信を削減する Chebyshev 基底共役勾配法 (CBCG 法) が提案されている[3][4]。

また、通信を削減する並列化モデルとして、MPI 並列とスレッド並列を組み合わせたハイブリッド並列がある。ハイブリッド並列は MPI 並列にスレッド並列を組み込むことで、MPI 並列よりも通信時間を削減できる[5]。

熊谷らの研究により、MPI 並列では CBCG 法の通信削減の効果があることが示されている[6]。本研究では、ハイブリッド並列で同様の実験を行うことにより、ハイブリッド並列下での CBCG 法の通信削減の効果を検証した。

## 2. Chebyshev 基底共役勾配法

CBCG 法の実装については参考文献[4]に基づい

Performance Evaluation of Chebyshev Basis Conjugate Gradient Method using Hybrid Parallel Programming Model  
Naoya Nomura<sup>†</sup>, Yosuke Kumagai<sup>†</sup>, Akihiro Fujii<sup>†</sup> and Teruo Tanaka<sup>†</sup>

<sup>†</sup>Kogakuin University

ている。CBCG 法の反復部のアルゴリズムを図1に示す。図1の6行目と9行目の行列積で集団通信が発生する。加えて、残差のノルムを計算するために内積計算が行われる。そのため、CBCG 法1反復につき3回の集団通信が行われる。

1	Repeat
2	$S_{n+1} = (T_0(A)r_n, T_1(A)r_n, \dots, T_k(A)r_n)$
3	If $n=0$
4	$Q_{n+1} = S_{n+1}$
5	Else
6	$B_{n+1} = (Q_n^T A Q_n)^{-1} Q_n^T A S_{n+1}$
7	$Q_{n+1} = S_{n+1} - Q_n B_{n+1}$
8	End If
9	$a_{n+1} = (Q_{n+1}^T A Q_{n+1})^{-1} Q_{n+1}^T r_n$
10	$x_{n+1} = x_n + Q_{n+1} a_{n+1}$
11	$r_{n+1} = r_{n+1} - A Q_{n+1} a_{n+1}$
12	$n = n + 1$
13	Until $\ r_n\ /\ b\  < \epsilon$

図1 CBCG 法の反復内部のアルゴリズム

## 3. ハイブリッド並列

ハイブリッド並列とは、分散メモリ型並列化手法 (MPI 並列) と共有メモリ型並列化手法 (スレッド並列) を組み合わせた並列化手法である。近年のスーパーコンピュータは、ノード内に複数コアがある SMP クラスタと呼ばれる形態がひとつの主要形態となっており、そのような形態の場合はハイブリッド並列が有効である。異なるノードのコア間には MPI 通信を用い、ノード内のコア間はスレッド並列で共有メモリを扱うことで、フラット MPI よりも通信を削減することができる。

## 4. 数値実験と評価

## 4.1 実験環境

本研究では、東京大学の FX10 スーパーコンピュータシステム (Oakleaf-FX) [7]を使用し、数値実験を行った。FX10 の構成を表1に示す。最大1,152ノードを使用し、1ノードに1プロセスを立ち上げ、16スレッド起動した。また、MPI 並列に

は MPI を用い、スレッド並列には OpenMP を用いてハイブリッド並列化を行った。実験ではサイズ  $144 \times 144 \times 144$  の 3 次元 Poisson 方程式のダルシー流れ問題から導出される拡散係数が不連続に変化する問題[8]を使用した。また、コア数を変化させても問題サイズが一定であるストロング・スケールリングとして実験した。

実験対象は CG 法と CBCG 法 ( $k=10, 20$ ) の 3 種類である。反復の終了条件は相対残差が  $1.0 \times 10^{-12}$  とした。

表 1 FX10 の構成

ノード	CPU 数	1
	メモリ	32GB
CPU	SPARK64™Ixfx	
	コア数	16 コア/CPU
	動作周波数	1.848GHz

#### 4.2 実験結果

表 2 に各解法の反復回数と集団通信回数を示す。表の反復回数の値は CG 法換算となっている。つまり、CBCG 法の項には実際の反復回数に  $k$  倍した数値を記載している。

コア数を増加させたときの実行時間を図 3 に示す。各手法ともコア数増加とともに実行時間は減少した。

コア数を増加させたときの通信時間を図 4 に示す。なお通信時間の計測には、集団通信の前に全プロセスで同期をとった。各手法とも 6,144 コアまでは通信時間は減少傾向にある。CG 法は 6,144 コアから増加傾向となった。一方 CBCG 法は 6,144 コアからあまり増加しなかった。これは、CBCG 法の通信削減効果が現れたためと考えられる。

表 2 反復回数と集団通信の回数

	反復回数	集団通信
CG	3,696	7,392
CBCG(10)	3,700	1,110
CBCG(20)	3,740	561

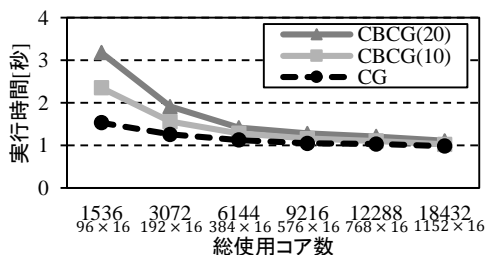


図 3 コア数増加による実行時間の推移

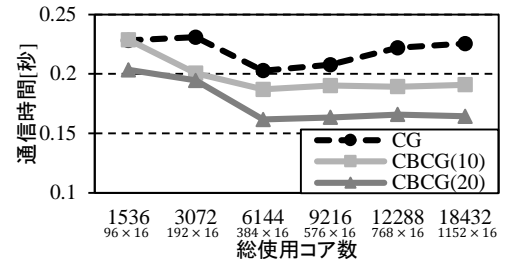


図 4 コア数増加による通信時間の推移

#### 5. おわりに

本研究では CBCG 法にハイブリッド並列を施し、高並列環境でハイブリッド並列を施した CG 法と比較することによりハイブリッド並列下での CBCG 法の性能評価を行った。CBCG 法の通信時間は CG 法よりも少なくなったことから、CBCG 法の通信削減効果はハイブリッド並列下でも現れることがわかった。しかし全体実行時間については CG 法もハイブリッド並列による通信削減の効果を受けているため、今回行った計測の最大コア数 18,432 コアでも通信削減効果が十分でなく CG 法が速いという結果となった。CBCG 法の実行時間は 18,432 コアを使用してもいまだ減少傾向にあるため、より高並列な環境下でも実行時間の減少が予想できる。今後の課題として、より高並列な環境で計測を行い、ハイブリッド並列を施した CBCG 法の実行時間がより多くのコア数を用いても減少していくかを検証する必要がある。

#### 参考文献

- [1] TOP500, <http://www.top500.org/lists/>.
- [2] Magnus R. Hestenes and Eduard Stiefel, Methods of Conjugate Gradients for Solving Linear Systems, Journal of Research of the Nation Bureau of Standards, Vol.49, pp.409-436(1952).
- [3] Mark Hoemmen, Communication-Avoiding Krylov Subspace Methods, PhD Thesis, University of California, Berkeley(2010).
- [4] 須田礼仁, 李聡, 島根浩平, 数値的に安定性な通信削減クロフ部分空間法, 計算工学講演会論文集, Vol.19(2014).
- [5] K.Nakajima, OpenMP/MPI Hybrid vs. Flat MPI on the Earth Simulator: Parallel Iterative Solvers for Finite Element Method, Lecture Notes in Computer Science 2858, pp.486-499(2003).
- [6] 熊谷洋佑, 藤井昭宏, 田中輝雄, 須田礼仁, 超高並列環境での通信削減を目的とした Chebyshev 基底共役勾配法の特性評価, 情報処理学会研究報告, Vol.2014-HPC-145, No.17(2014).
- [7] 東京大学情報基盤センタースーパーコンピュータシステム (Oakleaf-fx), <http://www.cc.u-tokyo.ac.jp/system/ox10/>.
- [8] Deutsch, C.V., Journel, A.G., GSLIB Geostatistical Software Library and User's Guide, Second Edition, Oxford University Press(1998).