

## AMG ライブラリの MPI/OpenMP ハイブリッド並列による高速化

水上忠彦<sup>†</sup> 藤井昭宏<sup>†</sup> 小柳義夫<sup>†</sup>

本研究では、MPI/OpenMP ハイブリッド並列モデルを AMG 法に適用させた。領域分割により MPI 化し、マルチカラーガウスザイデル法により OpenMP 化を行った。特に、Reordering によるメモリアクセスの効率化、First Touch、NUMA コントロールの三つの組み合わせを最適化し高速化を図った。評価には T2K オープンスパコン（東大）を利用し、評価した結果、MPI/OpenMP ハイブリッド並列モデルが FlatMPI のピーク性能の 1.5 倍程度の性能を示し、並列度によらず、Flat MPI と同等かそれ以上の性能を発揮することが分かった。

### Acceleration of Algebraic Multigrid Solvers library using MPI/OpenMP Hybrid parallelization

Tadahiko Mizukami<sup>†</sup> and Akihiro Fujii<sup>†</sup> and Yoshio  
Oyanagi<sup>†</sup>

We applied and evaluated MPI/OpenMP hybrid programming model for Algebraic Multigrid (AMG) method on T2K Open Supercomputer (Tokyo). We used domain decomposition for MPI and multi-color Gauss-Seidel method for OpenMP. Our implementation uses Re-ordering, First touch, and NUMA control for better performance. Numerical tests show that the hybrid model of AMG solver 1.5 times faster than flat MPI model, and that optimized version of the hybrid model AMG solver is faster than flat MPI model for all parallelism.

### 1. はじめに

近年、マルチコアプロセッサの普及に伴い、共有メモリ型並列計算機をネットワークで繋いだ分散・共有メモリ型並列計算機が広く用いられるようになった。このようなアーキテクチャの特性を活かすために、分散メモリ型の並列化手法と共有メモリ型の並列化手法を組み合わせたハイブリッド並列化手法が用いられている。そのため、現在ハイブリッド並列化手法は注目を集め、それらに関する研究が盛んとなっている。ハイブリッド並列化手法とは、一般的には MPI と OpenMP を組み合わせたスタイルである。中島の研究[1]では、MPI/OpenMP ハイブリッド並列モデルを不完全コレスキー分解を緩和法として用いた多重格子法に適用し、First Touch や NUMA コントロールなどの最適化について評価を行っている。

本研究では、MPI/OpenMP ハイブリッド並列モデルを、マルチカラーガウスザイデルを緩和法として使用し代数的多重格子法（AMG 法）に適用した。AMG 法をハイブリッド化するために、藤井が開発した MPI 並列版 AMG ライブラリ[2]を用い、各レベルで使用されているガウスザイデル法をマルチカラー法で並列性を抽出し OpenMP 化した。そして Reordering によるメモリアクセスの効率化、First Touch や NUMA コントロールによるメモリ配置の最適化を行い、T2K オープンスパコン（東大）上の最大で 8 ノード（128 コア）までを使用して、Flat MPI と MPI/OpenMP ハイブリッド並列モデルの評価を行った。結果、問題サイズを固定して評価すると、ハイブリッド並列モデルを適用し最適化を行うことで、Flat MPI のピーク性能に対して 1.5 倍程度の高速化をすることができた。

以下、2 章で AMG 法について簡単に説明し、3 章で使用したアプリケーションとハイブリッド化及び最適化について説明、4 章で計算環境、5 章で計測条件と結果、6 章でまとめる。

### 2. AMG 法

AMG 法は多重格子法（MG 法）の一種であり、問題  $Ax=b$  を解く際により小さいサイズの問題を生成し、それを利用して効率よく解く手法である。AMG 法は大きく分けて二つのプロセスからなる。一つは問題行列から未知数間のグラフ構造を作り、次のレベルに残す未知数を選択して粗いレベルを作る階層構造生成部である。問題を生成する部分を再帰的に行うことで複数のレベルの問題を生成することができる。二つ

<sup>†</sup> 工学院大学  
Kogakuin University (Japan)

目は生成された問題を使って実際に解く反復解法部である。反復解法部では上のレベルでガウスザイデル等の緩和法を行い、誤差を下レベルへと移し (restriction), 下のレベルで補正解を計算し元のレベルへと補間 (prolongation) する。図1はAMG法の概略を表したものである。

補間演算子を決めることで階層構造生成や各レベル間の移動が実現されるが、この補間演算子の作成の仕方によって様々なAMG法が考案される。

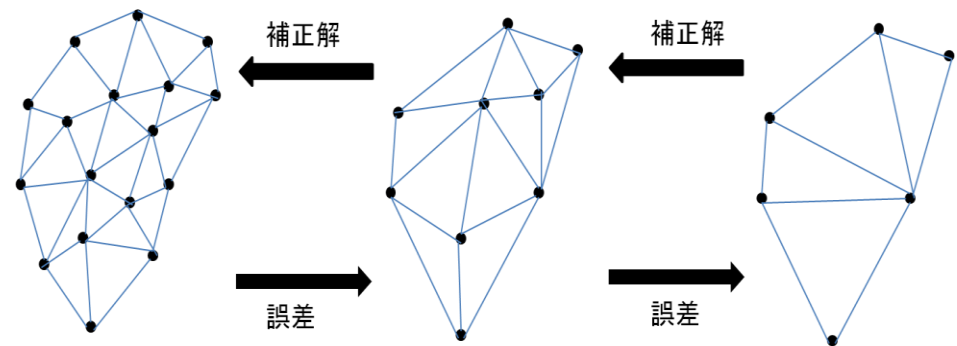


図1 代数的多重格子法の概略

本研究では Smoothed Aggregation に基づく AMG(SA-AMG)法[3]を用いた。SA-AMG法では問題行列から未知数間の依存関係を定義することができ、依存関係のある未知数同士で集合を作り、その集合内で重みづけをして補間演算子を生成する。SA-AMG法は様々な分野で利用されており、AMG法の代表的な手法の一つとなっている。

### 3. アプリケーション、ハイブリッド化実装

#### 3.1 AMGライブラリ

藤井が開発した AMG ライブラリは Fortran90 で書かれており、逐次版と MPI 版がある。本研究では MPI 版を使用した。AMG ライブラリでは各レベルでの緩和法を対称ガウスザイデル、レベル間の移動にはVサイクルを使用している。MPI 版では ParMETIS[4]というライブラリを使い、通信テーブルが極小化されるよう領域分割を行っている。ParMETIS とはグラフ分割や疎行列のオーダリングのための様々なアルゴリズムを持つ MPI ベースの並列ライブラリである。

#### 3.2 マルチカラー法

MPI/OpenMP ハイブリッド並列モデルで並列化を行う場合、領域分割された各領域に MPI プロセスを割り当て、各領域内で OpenMP による並列化が行われる。各領域においてガウスザイデルをそのまま OpenMP 化するとスレッド間でデータの競合が発生してしまい、計算が不安定となる。そこでマルチカラー法による色分けを行い、互いに独立な要素を見つけ出し並列性を抽出する (図2)。

本研究で使用した色分けアルゴリズムは以下である。

- ① パレットに色を1色用意する
- ② 問題行列の1行目に色を付ける
- ③ 2行目以降は隣接ノードの色を調べ、パレットに隣接ノードと異なる色がある場合はその色を付ける。無ければパレットに色を追加し、新しい色を付ける。

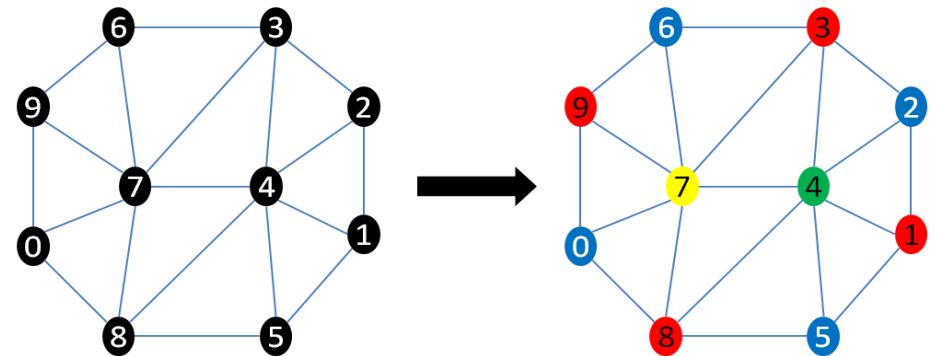


図2 ノードの色分けの例

#### 3.3 最適化

色分けされた問題の同一の色に属する要素は互いに独立であり、並列に計算する事が可能であるが、このままだと各スレッドは不連続なアクセスとなり効率が悪い。図3に示すように不連続となっている要素を色ごとにまとめ、連続となるようにデータをリオーダーリングした。

更に、First Touch[5]によるメモリ配置の最適化を行い、性能向上を図った。NUMAアーキテクチャでは、プログラムにおいて変数や配列を宣言した時点で物理メモリ上に領域は確保されない。ある変数に対してデータを格納した時点で物理的なメモリ上の場所が確定し、最初にアクセスしたコアと同じソケットのローカルメモリ上に領域

が確保される。First Touch とは、配列を初期化する時、各スレッドに実際の計算手順と同じ様に初期化をさせることである。これによって、データは各々のスレッドのローカルメモリに確保されるため大幅な速度向上が見込める場合もある。また NUMA コントロールを使用して、実行時のコア（またはソケット）とメモリを指定することで性能向上を図った。MPI/OpenMP ハイブリッド並列モデルにおいて、実行時に適切な NUMA コントロールを使用することによって、性能が向上することは既に明らかとなっている。[1]

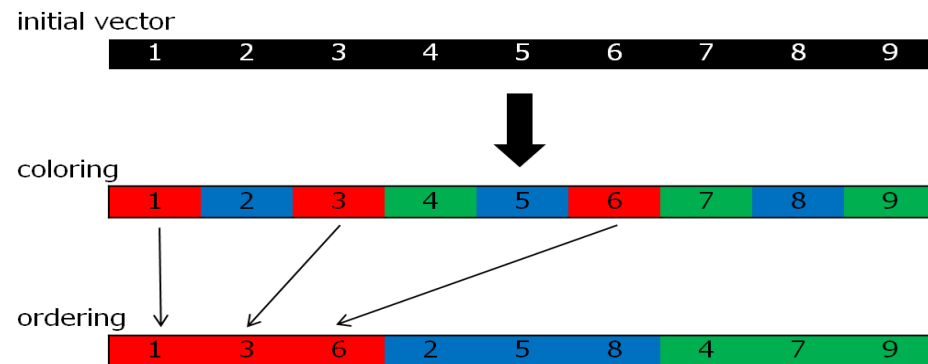


図 3 連続なデータアクセスのための並べ替え

#### 4. 計算環境

本研究では、T2K オープンスパコン（東大）[6]の 8 ノード（128 コア）までを使用して評価した。T2K（東大）は 16 個のプロセッサを持つ計算機 952 台を高速ネットワークで接続したクラスタ型並列計算機である。各ノードは NUMA アーキテクチャに基づき、AMD Quad Core Opteron 8356（2.3GHz）を 4 ソケット、合計 16 コアから構成されている（図 4）。ノードあたりの記憶容量は 32GB（一部 128GB）である。

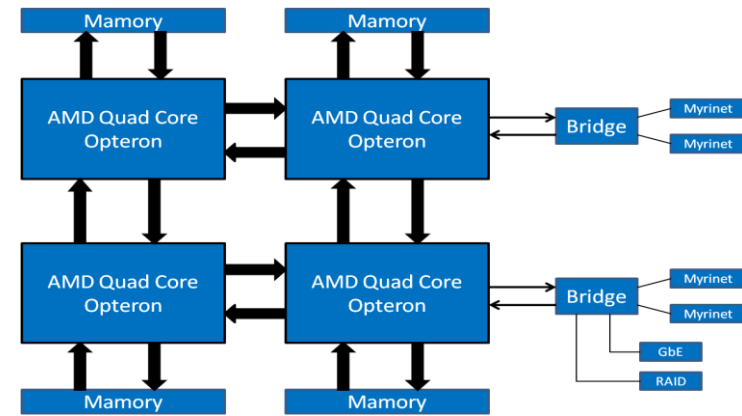


図 4 ノード構成

#### 5. 計算結果

##### 5.1 計測条件

本研究で行った最適化の効果について T2K（東大）を用いて評価した。使用した問題は 3 次元ポアソン方程式の等方性、問題サイズは small ( $50^3$  ( $50 \times 50 \times 50$ )), large ( $100^3$ ) の 2 つを扱った。計測では問題サイズを固定し、少しずつ並列度を上げていき実行時間を計測した。AMG 法の解法部では、マルチレベルのレベル間を移動することになるが、各レベルでは緩和法を 2 回適用し、最も粗いレベルでは 30 回適用している。1 反復内でのレベル間の移動には V サイクルを使用した。

以下の 3 種類の MPI/OpenMP ハイブリッド並列モデルと MPI 並列のみの Flat MPI を比較した。

Hybrid-2: スレッド数を 2 で固定し、MPI プロセス数を増やしていく。1 ノード内での MPI プロセス数は最大 8.

Hybrid-4: スレッド数を 4 で固定し、MPI プロセス数を増やしていく。1 ノード内での MPI プロセス数は最大 4.

Hybrid-8: スレッド数を 8 で固定し、MPI プロセス数を増やしていく。1 ノード内での MPI プロセス数は最大 2.

### 5.2 結果

2つの問題に対し、それぞれの並列モデルで計測を行い、計算性能を以下に示す(図5, 図6)。まず図5より問題サイズが小さい場合、Flat MPI (図中では単にMPIと表記)では通信のオーバーヘッド増加のため、32並列をピークとしてそれ以降は性能が低下している。一方ハイブリッド並列モデルでは、OpenMP化をしたことで領域数が減り通信のオーバーヘッドが少なくなり最適化前の状態でも高並列時にFlat MPIを上回る性能を発揮したが、低並列時にはFlat MPIに及ばなかった。しかし最適化を行うことで性能が改善され、低並列時でもFlat MPIと同等の性能を実現できた。

次に図6より問題サイズが大きい場合、ハイブリッド並列モデルは最適化前の状態では完全にFlat MPIに劣っているが、最適化後では大幅に性能が改善され特に高並列時における速度向上は顕著である。更に、Flat MPIでは128並列でピークとなっているが、ハイブリッド並列モデルは64並列でFlat MPIのピークと同等の性能を出せている。

ハイブリッド並列モデルを最適化することでFlat MPIのピーク時性能に対して

- ・問題サイズが小さい場合、最大で1.7倍
- ・問題サイズが大きい場合、最大で1.5倍

の性能が出ている。そしてどちらの問題に対しても hybrid-4 モデルが最適な結果となっている。

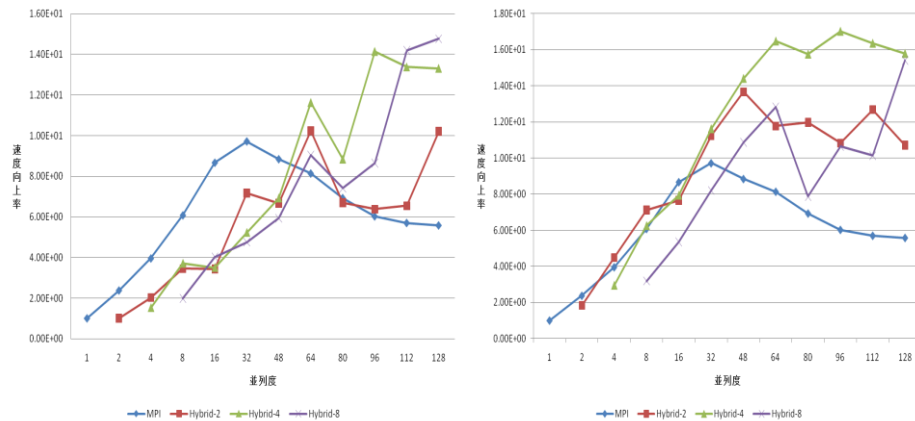


図5 small (50<sup>3</sup>) を解いた時の計算性能 (縦軸は Flat MPI を逐次で実行したときの計測時間を基準とした速度向上率, 横軸は使用したコア数) (左は最適化前, 右は最適化後 (NUMA コントロール, First Touch, 連続なメモリアクセスのためのリオーダーリング))

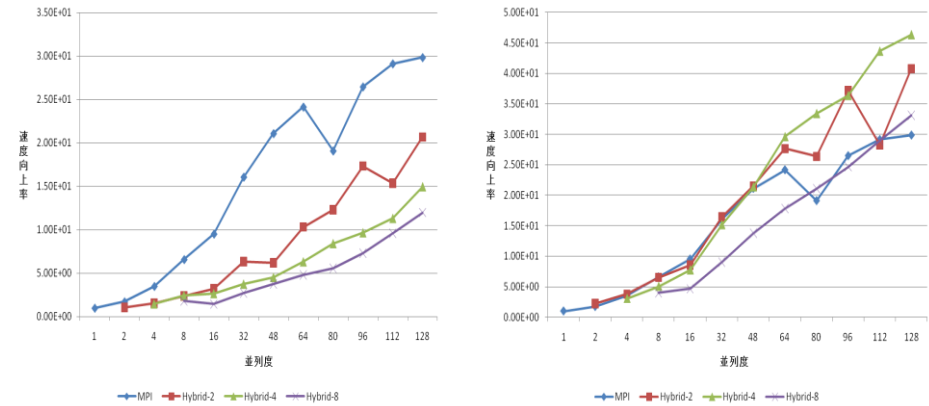


図6 large (100<sup>3</sup>) を解いた時の計算性能 (縦軸は Flat MPI を逐次で実行したときの計測時間を基準とした速度向上率, 横軸は使用したコア数) (左は最適化前, 右は最適化後)

### Small (50<sup>3</sup>)

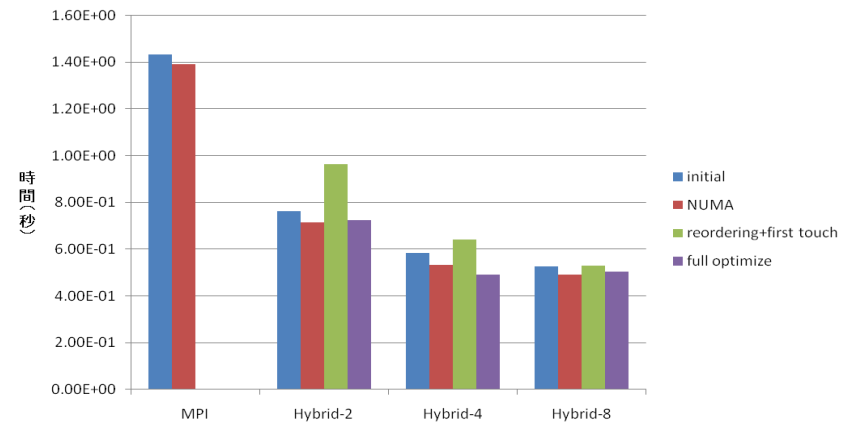


図7 Small (50<sup>3</sup>) ,128 並列時の各並列モデルの計算性能と最適化の効果 (■ initial (初期状態), ■ NUMA (NUMA コントロール), ■ reordering + first touch (連続データアクセスのための並び替え, First Touch), ■ full optimize (NUMA コントロール, 連続メモリアクセスのためのリオーダーリング, First Touch))

Large (100<sup>3</sup>)

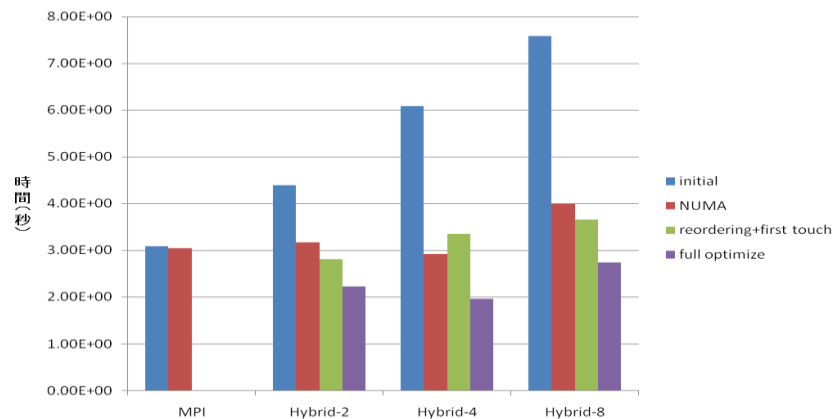


図 8 large (100<sup>3</sup>) 128 並列時の各並列モデルの計算性能と最適化の効果 (■initial (初期状態), ■NUMA (NUMA コントロール), ■reordering + first touch (連続データアクセスのための並び替え, First Touch), ■full optimize (NUMA コントロール, 連続メモリアccessのためのリオーダーリング, First Touch))

図 7 および図 8 は, 128 コア使用時において, 最適化の効果を各並列モデルについて比較したものである. 問題サイズが大きい場合 (図 8) において, ハイブリッド並列モデルは NUMA コントロールだけでも 2 倍近い性能向上が見られた. 更に First Touch と連続メモリアccessのためのリオーダーリングを適用することで Flat MPI 以上の性能を出すことができた.

問題サイズの小さい場合 (図 7) では, 最適化による効果は少ないように見える. これは 1 プロセスあたりが持つデータサイズが小さく, キャッシュに収まっていたために最適化の効果があまり得られなかったのではないかと考えられる.

## 6. まとめ

本研究では AMG 法に MPI/OpenMP ハイブリッド並列モデルを適用させ, 各レベルでの緩和法にはマルチカラーガウスザイデル法を使用した. そして連続メモリアccessのためのリオーダーリング, First Touch や NUMA コントロールによるメモリ配置の最

適化を行い, 問題サイズを固定して, T2K オープンスパコン (東大) の 128 コアまでを使用し評価を行った. 結果, MPI/OpenMP ハイブリッド並列モデルはスレッドを立てることで MPI プロセスを少なくすることができるため, 通信のオーバーヘッドによる性能低下を抑えることができ, Flat MPI と比べるとスケーラビリティが高いということがわかった. 問題サイズが小さい場合において, ハイブリッド並列モデルは最適化を行うことで, Flat MPI のピーク性能の 1.7 倍の性能であった. 問題が大きい場合においては, ハイブリッド並列モデルは Flat MPI のピーク性能の 1.5 倍の性能を達成している. 連続メモリアccessのためのリオーダーリング, First Touch や NUMA コントロールによるメモリ配置の最適化を行うことで, どの並列度のときでもハイブリッド並列モデルが Flat MPI と同等かそれ以上の性能を発揮することがわかった.

## 参考文献

- 1) 中島 研吾. OpenMP/MPI ハイブリッド並列プログラミングの多重格子法への適用. 出版地不明: 情報処理学会研究報告, Vol.2010-HPC-124 No.7, 2010.
- 2) 藤井昭宏, 小柳義夫. 科学技術シミュレーションにて多用される代数的多重格子法の評価. 出版地不明: シミュレーション 第 28 巻第 4 号, pp.9-14, 2009.
- 3) AMGS ライブラリ: <http://hpcl.info.kogakuin.ac.jp/olab/software>
- 4) P.Vanek and M.Brezina and J.Mandel. Algebraic Multigrid by Smoothed Aggregation for Second and Fourth Order Elliptic Problems: Technical Report UCD-CCM-036. (1995).
- 5) G.Karypis. ParMETIS: Parallel Graph Partitioning and Fill-reducing Matrix Ordering. Available from: <http://glaros.dtc.umn.edu/gkhome/metis/parmetis/downloadet>
- 6) 岩下武史, OpenMP 発展: <http://ais.sys.i.kyoto-u.ac.jp/~iwashita/openmp-advance-1.pdf>
- 7) 東京大学情報基盤センタースーパーコンピューティング部門: <http://www.cc.u-tokyo.ac.jp/>