階層型 CGA 法による並列多重格子法

中 島 研 吾

東京大学情報基盤センター

1. はじめに

本稿では，2013 年 5 月に実施された大規模 HPC チャレンジの結果を報告する．

著者等は，多重格子法（multigrid）による前処理手法を適用した並列反復法（MGCG 法）による連立一次方程式ソルバーに OpenMP/MPI ハイブリッド並列プログラミングを適用し，様々な評価を実施してきた〔1,2,3〕．〔1〕では，各 MPI プロセスにおける格子数が 1 となる粗い格子レベルでの修正方程式の解法（Coarse Grid Solver）を改良し，並列性能を大幅に改善することができた．〔2,3〕では，更なる安定化，高速化のために Coarse Grid Aggregation（CGA）を提案し，T2K オープンスパコン（東大）512 ノード（8,192 コア），富士通 PRIMEHPC FX10（Oakleaf-FX）4,096 ノード（65,536 ノード）を利用して検証した．〔3〕では更に疎行列格納法を従来の CRS（Compressed Row Storage）から ELL（Ellpack-Itpack）に変更した．ELL-CGA による MGCG 法は CRS 法によるオリジナル手法〔1〕と比較して，Oakleaf-FX 4,096 ノードを使用して弱スケーリングで 13%～35%，強スケーリングで 40%～70%の性能改善を得ることができた．

本稿では，CGA を改良した階層型 CGA 法（Hierarchical CGA，*h*CGA）を提案し，その効果について，Oakleaf-FX 4,096 ノードを使用した評価した結果を紹介する．

2. アプリケーション，実装

2.1 三次元地下水流れ問題シミュレーション

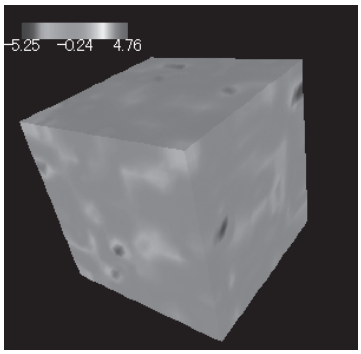
本研究では，図 1 に示すような不均質な多孔質媒体中の三次元地下水流れを並列有限体積法（Finite Volume Method，FVM）によって解くアプリケーションを扱う．対象とする問題は以下に示すような，ポアソン方程式および境界条件である：

$$\nabla \cdot \left(\lambda(x,y,z) \nabla \phi \right) = q, \phi = 0 \, at \, z = z_{\max}$$

ここで，ϕ は水頭ポテンシャル，$\lambda(x,y,z)$ は透水係数で位置座標の関数であり，セル（cell）ごとに異なっている．透水係数は，地質統計学の分野で使用される Sequential Gaussian アルゴリズム〔4〕により発生させた値を使用した（図 1（a））．q は体積フラックスであり，本研究では一様（=1.0）に設定されている．

透水係数の最小値，最大値，平均値はそれぞれ 10^{-5}，10^{+5}，10^{0} となるように設定されている．有限体積セルは一辺長さ 1.0 の立方体である．このような問題設定では，条件数が 10^{10} のオーダーとなるような対称，正定な悪条件マトリクスを係数とする線形方程式を解く必要がある．本研究で対象とするモデルは，各々 128^3 セルから構成される同じ不均質場に基づく部分モデルの集合である．したがって，x，y，z 各方向に周期的に同じ不均質パターンが繰り返される．

(a)



(b)

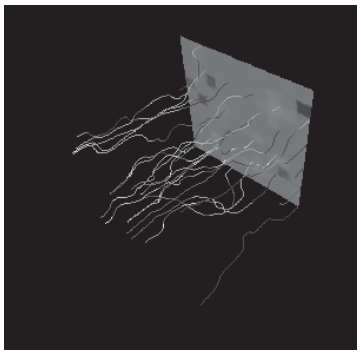


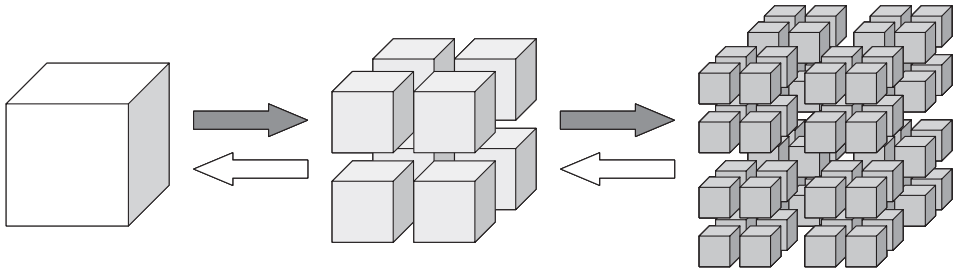
図 1　不均質多孔質媒体中の地下水流れの例（a）透水係数分布，（b）流線

2.2 多重格子法による前処理付き反復法

本研究では, ポアソン方程式を有限体積法によって離散化して得られる対称, 正定（Symmetric Positive Definite，SPD）な疎行列を係数行列とする連立一次方程式を，多重格子法（Mulrigrid）による前処理を施した共役勾配法（Conjugate Gradient Method，CG）によって解く．このような前処理付き共役勾配法を MGCG 法〔5〕と呼ぶ．残差ノルム|{b}-[A]{x}|/|b| が 10^{-12} 未満となるまで反復が繰り返される．

多重格子法は大規模問題向けのスケーラブルな解法として注目されている．Gauss-Seidel 法などの古典的反復法はセルサイズに相当する波長をもった誤差成分の減衰には適しているが，誤差の成分のうち，長い波長の成分は緩和を繰り返しても中々収束しない．多重格子法は，長い波長の成分が粗い格子上で効率的に減衰するという考え方に基づいている〔5〕．多重格子法は，細かい格子において対象とする線形方程式の残差を計算し，修正方程式を粗い格子へ補間（制限補間，Restriction）して解き，その結果を細かい格子に補間（延長補間，Prolongation）して誤差を補正するというプロセスを，再帰的に多段階に適用することによって構築可能である．各レベルの計算が適切に実施されれば，誤差のあらゆる長さの波長をもった成分を一様に減衰させることができるため，計算時間が問題規模に比例するいわゆる「Scalable」な手法の実現が可能である．本研究では，図 2 に示すように，8 個の「子（Children)」セルから 1 個の「親（Parent)」セルが生成されるような等方的な幾何学的多重格子法に基づき，格子間のオペレーションとしては，最密格子と最疎格子の間を直線的に動く V サイクル〔5〕を採用した．本研究では，各レベルにおける多重格子法のオペレーションは並列に実施されるが，〔1〕に示すオリジナル手法では最も粗い格子レベル（図 2 における Level=k）では 1 コアに集めて計算を実施する．従って最も粗い格子レベルでは，領域数（MPI プロセス数）＝格子数となる．

並列多重格子法では各レベルにおいて通信が必要となるが，粗い格子レベルでは，計算量が相対的に減少するため，領域間の通信，特に MPI の立ち上がりの Latency の効果を無視できなくなる．大規模な計算機システムを用いて大規模な問題を解く場合には，レベル数が大きくなり，領域間通信に対する配慮が必要となる．このような場合に，MPI プロセス数を減らすことのできるハイブリッド並列プログラミングモデルは有効である．



Level= k　　Level= k-1　　Level= k-2

図 2　幾何学的多重格子法のプロセス（8 Children=1 Parent）

多重格子法では，各レベルにおける線形方程式を緩和的に計算するための演算子を緩和演算子（Smoothing Operator, Smoother）と呼んでいる. 緩和演算子として代表的なものは Gauss-Seidel 法であり多くの研究で使用されているが，悪条件問題向けには不完全 LU 分解，不完全コレスキー分解が有効である〔1,2,3,5〕. 本研究では，フィルインを生じない不完全コレスキー分解（IC(0)）を緩和演算子として採用した. IC(0)のプロセス（分解，前進後退代入）は大域的な処理を含むため，並列化は本来困難である. 各領域において独立に IC(0)処理を実施するような，ブロック Jacobi 型の局所処理によって並列化は可能であるが，特に悪条件問題の場合，領域数が増えると収束が悪化する. ここで, 加法シュワルツ法（Additive Schwartz Domain Decomposition, 以下 ASDD）〔6〕を組み合わせることにより，並列計算においても安定した解を得ることが可能となる. ASDD 法のアルゴリズムは以下の通りである：

① M を全体前処理行列，r と z をベクトルとして，$Mz=r$ を前進後退代入によって解くものとする.

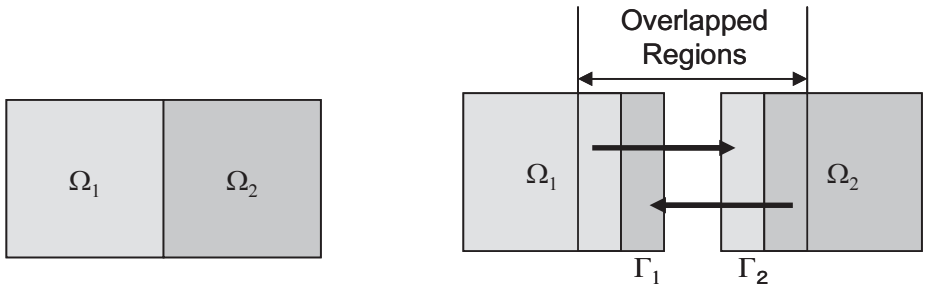
② 全体領域を図 3（a）に示すような 2 領域，すなわち，Ω_1 および Ω_2 に分割したと仮定し，各領域で独立に局所前処理を実施する：

$$z_{\Omega_1} = M_{\Omega_1}^{-1} r_{\Omega_1}, \quad z_{\Omega_2} = M_{\Omega_2}^{-1} r_{\Omega_2}$$

③ 各領域間のオーバーラップ領域 Γ_1 および Γ_2 の効果を次式によって導入する（図3(b)）. ここで n は ASDD のサイクル数である：

$$z_{\Omega_1}^{n} = z_{\Omega_1}^{n-1} + M_{\Omega_1}^{-1}(r_{\Omega_1} - M_{\Omega_1} z_{\Omega_1}^{n-1} - M_{\Gamma_1} z_{\Gamma_1}^{n-1}) \quad z_{\Omega_2}^{n} = z_{\Omega_2}^{n-1} + M_{\Omega_2}^{-1}(r_{\Omega_2} - M_{\Omega_2} z_{\Omega_2}^{n-1} - M_{\Gamma_2} z_{\Gamma_2}^{n-1})$$

④ ②，③を繰り返す.



(a) Local operations　　(b) Global nesting operations

図 3　加法シュワルツ法（Additive Schwartz Domain Decomposition，ASDD）

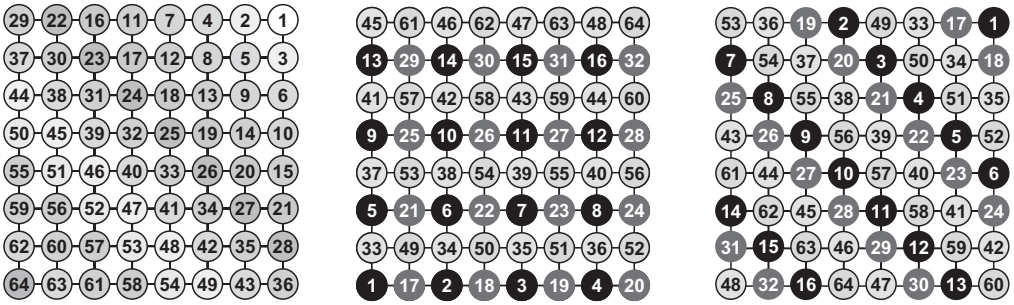
2.3 リオーダリング，最適化手法

OpenMP/MPI ハイブリッド並列プログラミングモデルで，FVM によるアプリケーションを並列化する場合，領域分割された各領域に MPI のプロセスが割り当てられ，各領域内で OpenMP による並列化が行われる．各領域においては，不完全コレスキー分解のように大域的な依存性を含むプロセスについては，各要素の並べ替え（Reordering）により依存性を排除し，並列性を抽出する手法が広く使用されている〔1,2,3〕．Hyper-Plane／Hyper-Line 法と類似した level-set に基づく Reverse Cuthill-McKee （RCM）法（図 4（a））はマルチカラー法（Multicoloring，MC）（図 4（b））と比較して，悪条件問題に対して安定であるが，各レベルにおける要素数が不均質となるため，並列性能が必ずしも高くない．本研究では，並列性が高く悪条件問題に対して安定な CM-RCM 法による並び替えを適用している〔7〕．本手法は，RCM 法の各レベルをサイクリックに再番号付けする Cyclic マルチカラー法（Cyclic Multicoloring，CM）を組み合わせたものである（図 4（c）参照）．CM-RCM 法では各「色」内の要素は独立で，並列に計算を実行することが可能である．CM-RCM 法の色数の最大値は RCM におけるレベル数の最大値である．本研究では多重格子法の各レベルにおいて CM-RCM 法を適用している．

CM-RCM 法による並べ替えでは，

① 同一の色（またはレベル）に属する要素は独立であり，並列に計算可能
② 「色」の順番に番号付け
③ 色内の要素を各スレッドに振り分ける

という方式〔1〕を採用しているが，同じスレッド（すなわち同じコア）に属する要素番号は連続では無いため，効率が低下する可能性がある．図 8 に示すように同じスレッドで処理するデータを連続に配置するように更に並び替え（Sequential Reordering）ることによって，色数の多い場合のキャッシュヒット率を高め，性能向上を図っている（図 5）〔1〕．



(a)RCM　(b) MC: 4 colors　(c) CM-RCM: 4 colors

図 4 MC（Multicoloring）, RCM（Reverse Cuthill-McKee）， CM-RCM による再番号付け例〔1,2,3,7〕

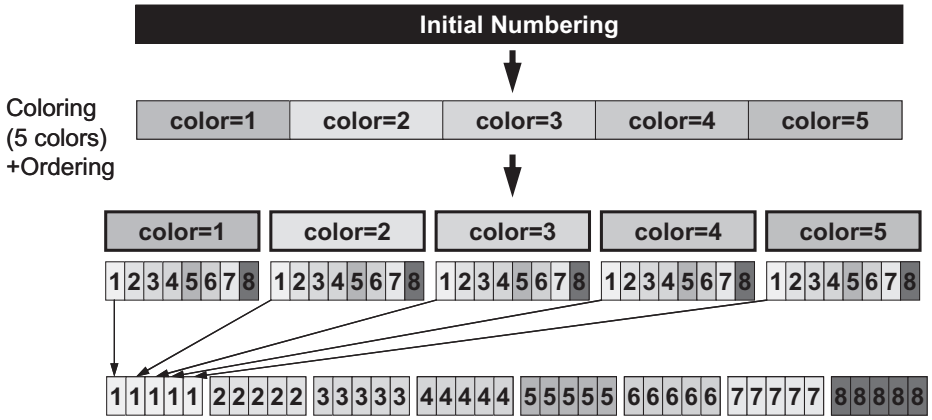


図 5 連続データアクセスのためのデータ再配置（sequential reordering）（5 色，8 スレッド）

2.4 ハイブリッド並列プログラミングモデル

本研究では，以下に示す 3 種類の OpenMP/MPI ハイブリッド並列プログラミングモデルを適用し，全コアに独立に MPI プロセスを発生させる Flat MPI と比較した．Oakleaf-FX の各ノードは 16 コアから構成されているため，各ノードにおける MPI プロセス数と各プロセスの OpenMP スレッド数の積が 16 となるように設定されている：

- **Hybrid 4×4（HB 4×4）**：各ノードにスレッド数 4 の MPI プロセスを 4 つ起動する
- **Hybrid 8×2（HB 8×2）**：各ノードにスレッド数 8 の MPI プロセスを 2 つ起動する
- **Hybrid 16×1（HB 16×1）**：1 ノード全体に 16 の OpenMP スレッド，1 ノード当たりの MPI プロセスは 1 つ

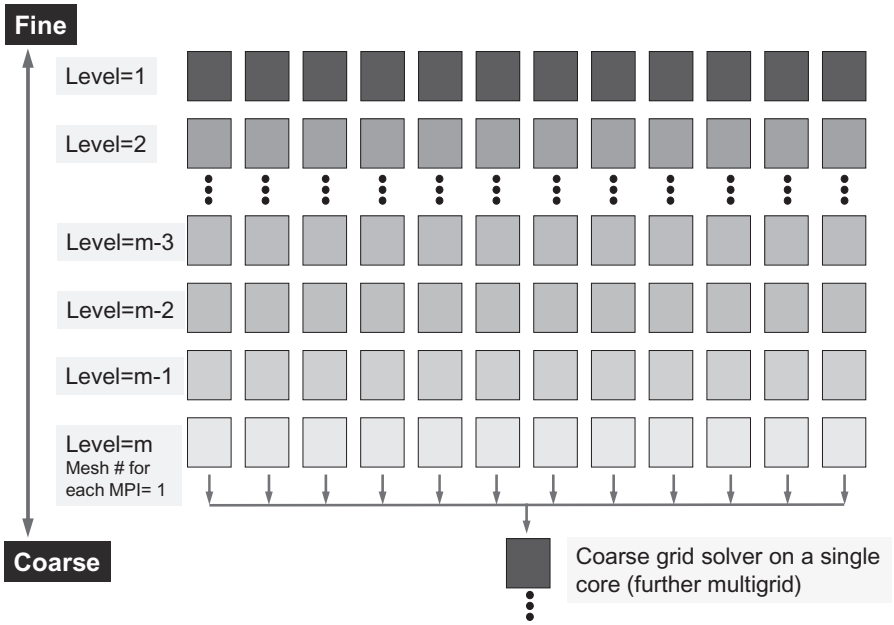


図 6　オリジナル Coarse Grid Solver〔1〕

3. 階層型CGA法（*h*CGA）

〔1〕に示すオリジナル手法では，各 MPI プロセスにおける格子数が 1 となる最も粗い格子レベルで 1 コアに集めて Coarse Grid Solver を適用している．Coarse Grid Solver の問題サイズは領域数（MPI プロセス数）と等しくなる．〔1〕では Coarse Grid Solver としてマルチグリッド（V-cycle）を適用し，収束するまで V-cycle を繰り返している（図 6）．

並列多重格子法では，MPI プロセス数が増加した場合，特に粗いレベルにおける通信によるオーバーヘッドによる低下が懸念されている．〔2,3〕では，粗いレベルにおけるプロセスを aggregate する CGA 法（Coarse Grid Aggregation）（図 7）により，通信によるオーバーヘッドを削減することを試みた．CGA 法では図 6 に示すオリジナル手法よりも細かい格子レベル（レベル数が少ない場合）で Coarse Grid Solver に移行する．オリジナル手法では Coarse Grid Solver は 1 コアで実行していたが，CGA 法では 1 MPI プロセスに集め，OpenMP によるマルチスレッド並列化を実施している．〔3〕では更に疎行列格納法を従来の CRS（Compressed Row Storage）から ELL（Ellpack-Itpack）に変更した．ELL-CGA による MGCG 法は CRS 法によるオリジナル手法〔1〕と比較して，Oakleaf-FX 4,096 ノードを使用して弱スケーリングで 13%～35%，強スケーリングで 40%～70% の性能改善を得ることができた．

図 8 は Oakleaf-FX 4,096 ノード（65,536 コア），HB　8×2 を使用した場合の， ELL 格納形式，CGA 法の効果である．メッシュ数は 17,179,869,184 である．コロン（:）の後の 2 桁整数値は収束までの反復解法である．CGA 法の適用により Coarse Grid Solver への移行がより細かい格子レベル（レベル数が少ない場合）で実施されるほど反復回数が減少している．CGA 法を適用すると，IC(0) スムージングが 1 MPI プロセス上で実行される割合が増えるため，より収束が安定化するためと考えられる〔2,3〕．格子レベル＝6 において Coarse Grid Solver へ移行した場合，反復回数は減少するが，Coarse Grid Solver の計算時間が増加するため，格子レベル＝7（最適値）で移行した場合と比較して全体の計算時間は長くなる．

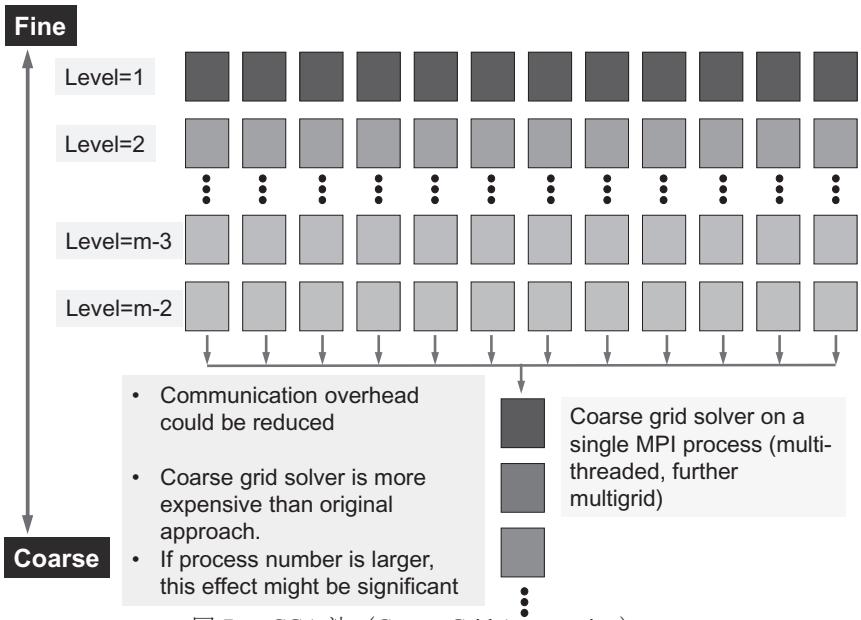


図 7　CGA 法（Coarse Grid Aggregation）

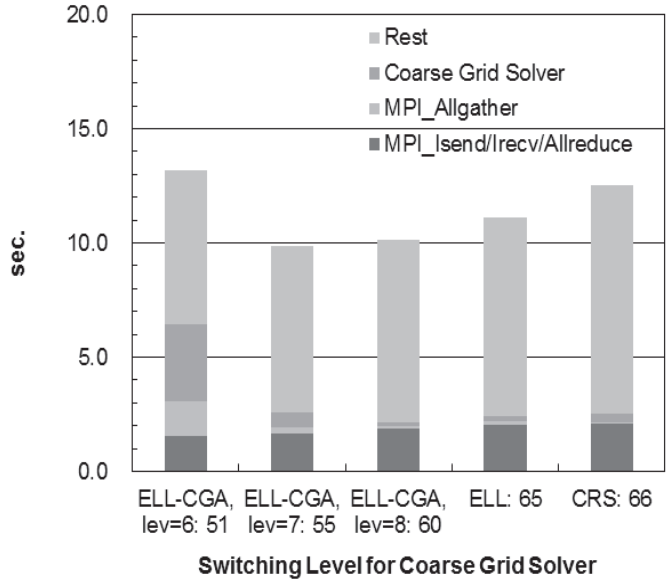


図 8　CGA 法の Oakleaf-FX 4,096 ノード（65,536 コア）での性能，MGCG 法の実行時間（HB 8×2，17,179,869,184 自由度）

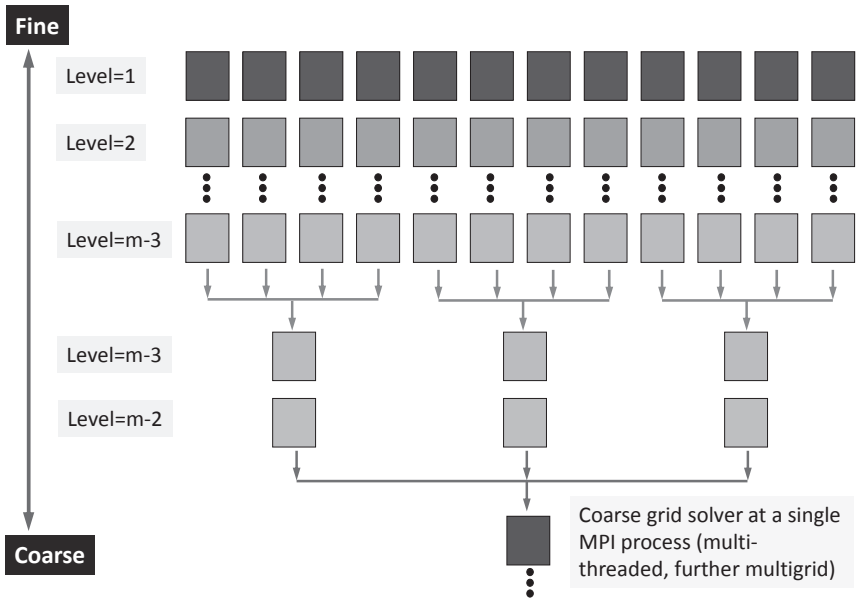


図 9　*h*CGA 法（Hierarchical Coarse Grid Aggregation）

本研究では，更に粗い格子レベルでの通信オーバーヘッドを回避するため，*h*CGA 法（Hierarchical Coarse Grid Aggregation，階層型 CGA 法）（図 9）を提案し，その評価を実施した．*h*CGA 法と従来の CGA 法との相違は 1MPI プロセスによる CGA 法への移行前に複数の MPI プロセスを統合し全体のプロセス数を減少させることにより，通信オーバーヘッドを削減することにある．

図 10 は図 8 に示したのと同じメッシュ数は 17,179,869,184 の問題を Oakleaf-FX 4,096 ノード（65,536 コア），HB 8×2 によって解いた場合の *h*CGA 法の効果である．格子レベル 6 で 4,096 ノ

ードから 128 ノードへ移行し，その後，格子レベル 7 で 1 MPI プロセス（8 コア）に移行することによって CGA 法の最適値と比較して計算時間が短縮されており，*h*CGA 法の有用性が示されている．

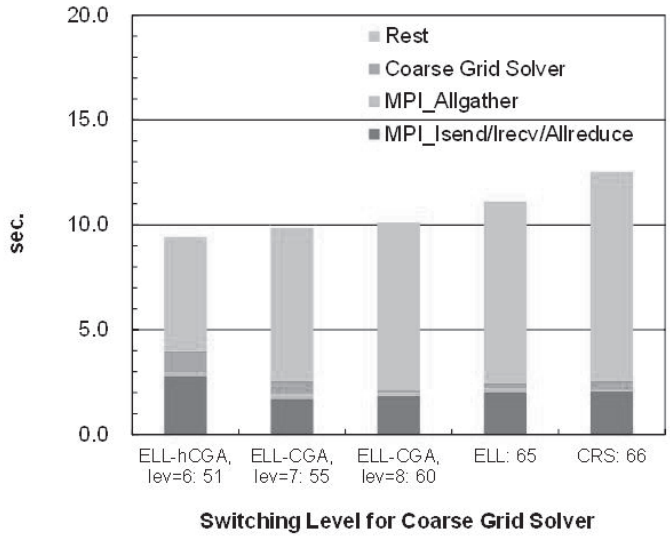


図 10　CGA 法，*h*CGA 法の Oakleaf-FX 4,096 ノード（65,536 コア）での性能，MGCG 法の実行時間（HB 8×2，17,179,869,184 自由度）

4. まとめ

本稿では，著者の提案した階層型 CGA 法（*h*CGA 法）を MGCG 法に適用し，Oakleaf-FX 4,096 ノードを使用した大規模問題（17,179,869,184 自由度）において，従来の CGA 法と比較して計算時間を短縮することができ，*h*CGA 法の有用性を確認できた．

今後は，MPI プロセス統合時のノード数，移行レベル，Coarse Grid Solver への移行レベルの最適値を自動的に決定する仕組みについて検討を実施するとともに，各演算，通信の更なる高速化を図る予定である．

参考文献

[1] Nakajima, K., New Strategy for Coarse Grid Solvers in Parallel Multigrid Methods using OpenMP/MPI Hybrid Programming Models, ACM Proceedings of PPoPP/PMAM 2012, New Orleans, LA, USA, ACM Digital Library (DOI: 10.1145/2141702.2141713), 2012

[2] Nakajima, K., OpenMP/MPI Hybrid Parallel Multigrid Method on Fujitsu FX10 Supercomputer System, IEEE Proceedings of 2012 International Conference on Cluster Computing Workshops, 199-206, IEEE Digital Library: 10.1109/ClusterW.2012.35, 2012

[3] Nakajima, K., Large-scale Simulations of 3D Groundwater Flow using Parallel Geometric Multigrid Method, Procedia Computer Science 18, 1265-1274, Proceedings of IHPCES 2013 (Third International Workshop on Advances in High-Performance Computational Earth Sciences: Applications and Frameworks) in conjunction with ICCS 2013, Barcelona, Spain, 2013

[4] Deutsch, C.V., Journel, A.G., GSLIB Geostatistical Software Library and User's Guide, Second Edition. Oxford University Press, 1998

[5] Tottemberg, U., Oosterlee, C. and Schuller, A., Multigrid, Academic Press, 2001

[6] Smith, B., Bjϕrstad, P. and Gropp, W., Domain Decomposition, Parallel Multilevel Methods for Elliptic Partial Differential Equations, Cambridge Press, 1996

[7] Washio, T., Maruyama, K., Osoda, T., Shimizu, F., Doi, S., Efficient implementations of block sparse matrix operations on shared memory vector machines. Proceedings of The 4th International Conference on Supercomputing in Nuclear Applications (SNA2000), 2000