数理解析研究所講究録
第1733巻 2011年 1-10

マルチコア時代の並列前処理手法

Parallel Preconditioning Methods for Iterative Solvers in Multi-Core Era

中島 研吾[1)2)]
Kengo NAKAJIMA

1) 東京大学情報基盤センター（〒113-8658 東京都文京区弥生2の11の16, nakajima@cc.u-tokyo.ac.jp）
2) 科学技術振興機構戦略的創造研究推進事業（CREST）

OpenMP/MPI hybrid parallel programming models were implemented to 3D finite-volume based simulation code for groundwater flow problems through heterogeneous porous media using parallel iterative solvers with multigrid preconditioning. Performance and robustness of the developed code has been evaluated on the "T2K Open Supercomputer (Tokyo)" and "Cray-XT4" using up to 8,192 cores through both of weak and strong scaling computations. OpenMP/MPI hybrid parallel programming model demonstrated better performance and robustness than flat MPI with large number of cores for ill-conditioned problems with appropriate command lines for NUMA control, first touch data placement, reordering of the data for contiguous "sequential" access to memory and appropriate coarse grid solver.

Key Words Multigrid, OpenMP/MPI Hybrid, Preconditioning

1. はじめに

近年，マルチコアプロセッサの普及，大規模システムにおけるコア数の増加を背景として，ハイブリッド(Hybrid)並列プログラミングモデルが脚光を浴びるようになり，Flat MPI（またはPure MPI）との優劣に関する議論が盛んとなっている（図1）．Hybrid並列プログラミングモデルはメッセージパッシングによる「Coarse-Grain Parallelism」と，ディレクティブによる「Fine-Grain Parallelism」の融合であり，一般的にはMPIとOpenMPを組み合わせたスタイルである．2018年頃に登場すると言われているExascale System（1秒間に10^{18}回の浮動小数点演算能力を持つ(Exa FLOPS)）は，億単位のコアから構成されるものと想定され，現在広く使用されているFlat MPIを適用することは困難と考えられており，MPIプロセス数を減らすことのできるHybrid並列プログラミングモデルへの期待は大きい．

著者は〔1〕において，有限要素法に基づく三次元弾性静力学問題向けシミュレーションで使用されている前処理つき反復法にOpenMP/MPIハイブリッド並列プログラミングモデルを適用し，T2Kオープンスパコン（東大）（以下「T2K（東大）」）〔2〕の512コアを使用した評価を実施した．OpenMP/MPIハイブリッド並列プログラミングモデルは適切なNUMA controlの組み合わせにより，OpenMP/MPIハイブリッド並列プログラミングモデルがFlat MPIと同等かそれを上回る性能を発揮することがわかった．更に，First Touch Data Placement，連続メモリアクセスのためのデータ再配置を適用することにより，特にコア当たり問題規模が小さい場合の性能が改善されることが明らかとなった．

本研究では，OpenMP/MPIハイブリッド並列プログラミングモデルを，〔3〕で開発された，並列多重格子前処理付き反復法を使用した，三次元有限体積法に基づく不均質場における地下水流れ問題シミュレーションに適用した．本研究ではT2K（東大）の他，米国ローレンスバークレイ国立研究所National Energy Research Scientific Computing Center（NERSC）の有する「Cray-XT4」〔4〕の8,192コアまでを使用して，Flat MPIとOpenMP/MPIハイブリッド並列プログラミングモデルの評価を実施した．

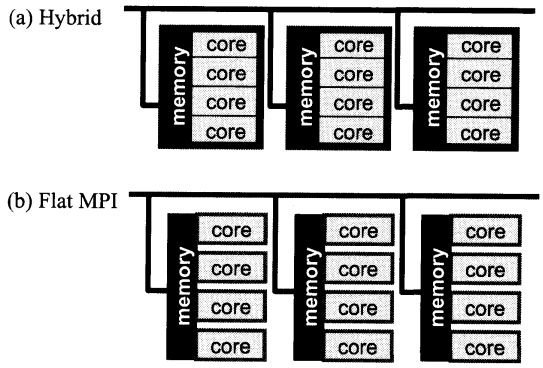


図1 並列プログラミングモデルの例

2. 計算環境

本研究では，T2Kオープンスパコン（東大）（T2K（東大））〔2〕，およびCray-XT4（NERSC，National Energy Research Scientific Computing Center）8,192コアまでを使用して評価した．

T2K（東大）は筑波大，東大，京大の3大学で定められた「T2Kオープンスパコン仕様」に基づき日立製作所が製作した952ノード，15,232コア，ピーク性能140 TFLOPSのクラスタ型コンピュータシステムである〔5〕．

各ノードはcc-NUMA（Cache-Coherent NUMA, Non Uniform Memory Access）アーキテクチャに基づきAMD Quad-Core Opteron（2.3GHz）4ソケット，合計16コアから

構成されている（図2）．コア当たりのピーク性能は9.2GFLOPSである．ノードあたりのピーク性能，記憶容量はそれぞれ147.2GFLOPS，32GB（一部128GB）である．

Cray-XT4（Franklin）システムは図2に示したAMD quad-core Opteron（2.3GHz）1ソケットを1ノードとしたクラスタ型コンピュータシステムであり，9,572ノード，38,288コア，ピーク性能352TFLOPSである．表1に両システムのネットワーク諸元を示す．ネットワークトポロジはT2K（東大）が多段クロスバーに対して，Cray-XT4は3Dトラス構造である．各ソケットのメモリ性能については，T2K（東大）はDDR2 667MHz，Cray-XT4はDDR2 800MHzであり，Cray-XT4の性能が高い．

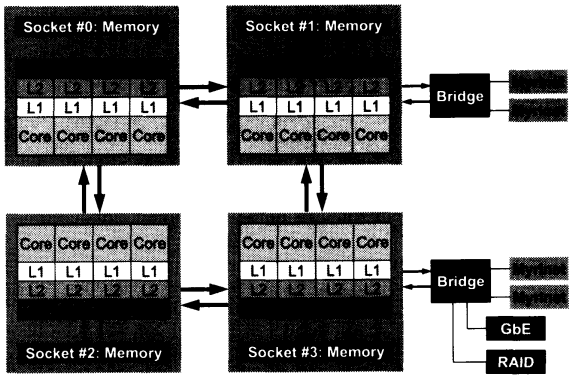


図2　T2K（東大）1ノード，各ノードはAMD Quad-core Opteron（2.3GHz）4つ搭載

表1　T2K（東大），Cray-XT4のノード概要

	T2K（東大）	Cray-XT4
Interconnect	Myrinet-10G×4	Cray SeaStar2
Network Topology	Multistage Crossbar	3D Truss
Comm. Bandwidth (GB/sec)	5.0	7.6
Comm. Latency (μsec)	2.0	5.0 (nearest neighbor) 6.0 (far-away nodes)

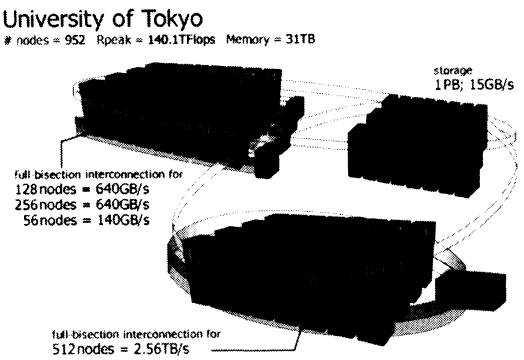


図3　T2K（東大）の概要〔2〕

T2K（東大）は図3に示すように，内部でクラスタ群に分かれている．各クラスタ群内のノード間はMyrinet-10G（1リンクあたり1.25GB/sec×双方向）で接続されている．ノードA群は各ノード4本（5.00GB/sec×双方向），ノードB群は2本（2.50GB/sec×双方向）である．本研究ではノードA群の512ノード（合計8,192コア）を使用した．

3. アプリケーション，実装

3.1 三次元地下水流れ問題シミュレーション

本研究では，図4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：

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

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〔6〕により発生させた値を使用した（図4（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．

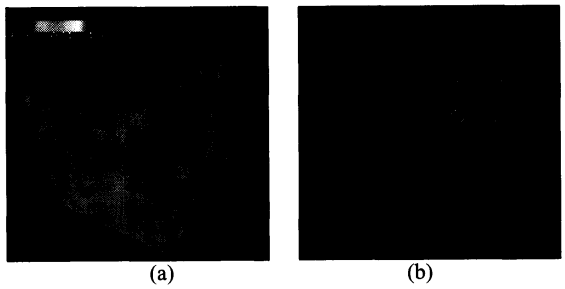


図4　不均質多孔質媒体中の地下水流れの例
（a）透水係数分布，（b）流線

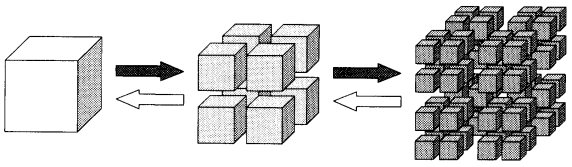
3.2 多重格子法による前処理付き反復法

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〔3〕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〔7〕．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 5 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〔7〕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，最も粗い格子レベル（図 5 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＝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 5　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〔3,7,8〕．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）〔9〕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 6 (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：

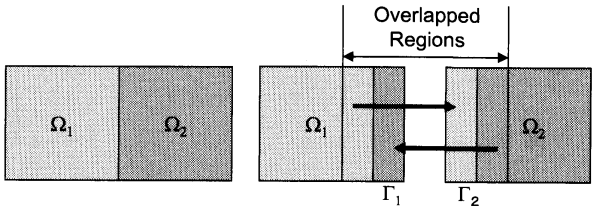
$$z_{\Omega_1} = M_{\Omega_1}^{-1} r_{\Omega_1},\quad z_{\Omega_2} = M_{\Omega_2}^{-1} r_{\Omega_2}$$

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 6 (b)）．ここで n は ASDD のサイクル数である：

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

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

④ ②，③を繰り返す．



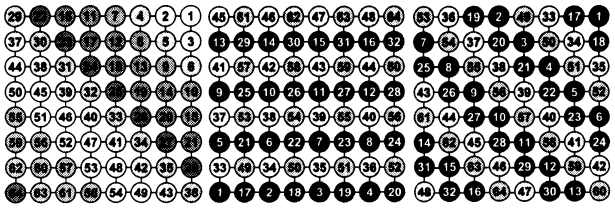
(a) Local operations　　(b) Global nesting operations

図 6　加法シュワルツ法（Additive Schwartz Domain Decomposition，ASDD）

3.3 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〕．Hyper-Plane／Hyper-Line 法と類似した level-set に基づく Reverse Cuthill-McKee（RCM）法（図 7）はマルチカラー法（Multicoloring，MC）（図 7）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〔1〕．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 7 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．



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

図 7 MC（Multicoloring），RCM（Reverse Cuthill-McKee），CM-RCM による再番号付け例〔1,10〕

T2K（東大）において OpenMP/MPI ハイブリッド並列プログラミングモデルを使用する場合，NUMA アーキテクチュアの特性を利用するための実行時制御コマンド（NUMA control）を使用して，コア（またはソケット）とメモリの関係を明示的に指定することによって，性能が向上することは既に明らかとなっている〔1〕．本研究では，様々な実行時制御コマンドの組み合わせの中で最適のものを選択して適用した．更に，

- First Touch Data Placement〔11〕の適用
- 連続データアクセスのためのデータ再配置（sequential reordering）

によって性能の改善を実施する〔1〕．NUMA アーキテクチュアでは，プログラムにおいて変数や配列を宣言した時点では，物理的なメモリ上に記憶領域は確保されず，ある変数を最初にアクセスしたコア（の属するソケット）のローカルメモリ上に，その変数の記憶領域が確保される．これを First Touch Data Placement〔11〕と呼び，配列の初期化手順により大幅な性能の向上が達成できる場合もある．具体的には，実際の計算の手順にしたがって配列を初期化することによって実現できる．

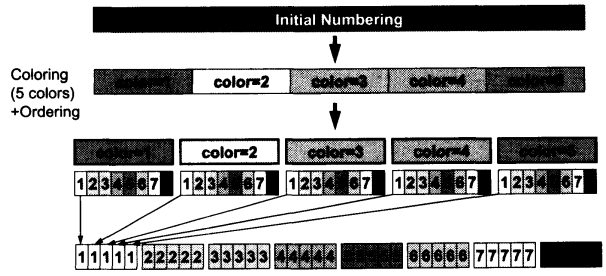


図 8　連続データアクセスのためのデータ再配置（sequential reordering）（5 色，8 スレッドの場合）

また，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），First Touch Data Placement を併用することによって性能向上を図る〔1〕．

4. 計算結果

4.1 最適化の効果

提案手法の安定性と効率について，T2K（東大），Cray-XT4 を使用して評価した．MGCG 法の多重格子法部分の緩和演算子としては IC(0)を適用し，V サイクルの各レベルにおいて 2 回の反復，また各反復において ASDD を 1 回適用した．CG 法の各反復において V サイクル 1 回を適用した．最も粗い格子における解法（Coarse Grid Solver）としては，各領域 1 セルになった状態（図 5 の Level=k）で 1 コアに集め，IC(0)スムージングを 2 回施す方法を適用している．

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：

- **Hybrid 4×4（HB 4×4）**：スレッド数 4 の MPI プロセスを 4 つ起動する，各ソケットに OpenMP スレッド 4，ノード当たり 4 つの MPI プロセス
- **Hybrid 8×2（HB 8×2）**：スレッド数 8 の MPI プロセスを 2 つ起動する，2 ソケットに OpenMP スレッド 8，ノード当たり 2 つの MPI プロセス（T2K（東大）のみ）
- **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（T2K（東大）のみ）

Cray-XT4 は 1 ノードあたり 1 ソケット（4 コア）であるため，上記のうち HB 4×4 のみを適用した．

まず，最初に 3.3 で述べたリオーダリング，最適化の効果を評価するため，64 コア（T2K（東大）：4 ノード，Cray-XT4：16 ノード）を使用した評価を実施した．各コアにおける問題サイズ（セル数）は 262,144（$=64^3$），全問題サイズは 16,777,216 である．図 9 は各並列プログラミングモデルにおける，収束までの反復回数と CM-RCM の色数の関係である．本研究で対象としているような，規則的

な差分格子形状では，CM-RCM 法において 2 色あれば並列計算を実施することが可能である．

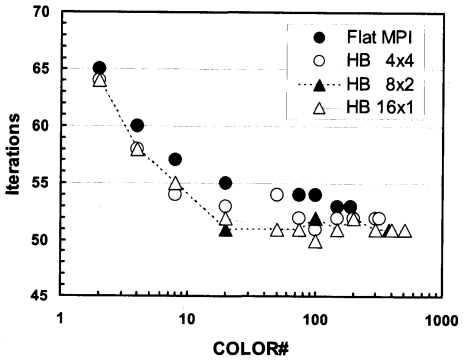


図 9　MGCG 法の反復回数と色数の関係(16,777,216 セル，64 コア)（不均質多孔質媒体中の三次元地下水流れ）

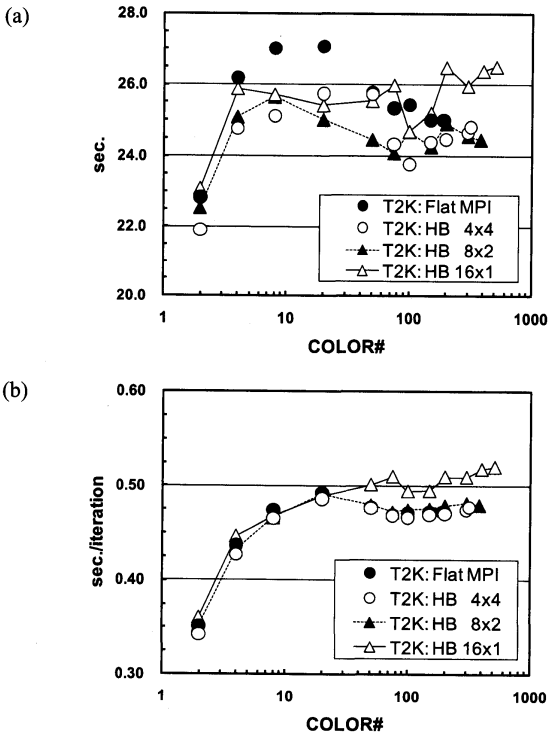


図 10　MGCG 法の計算性能と CM-RCM 法の色数の関係（(a) MGCG ソルバー計算時間，(b) 1 反復あたり計算時間）(16,777,216 セル，64 コア，T2K（東大)）(不均質多孔質媒体中の三次元地下水流れ）(3.3 に示した最適化（NUMA Control，First Touch Data Placement，連続データアクセスのための再データ配置（図 8）適用後))

一般に色数が増加すると Incompatible Nodes の数が減少するため反復回数は減少する〔12〕．本研究で対象とするような不均質な問題の場合，必ずしもその通りでは無いが，図 9 に示すように，一般的傾向としては色数の現象とともに，反復回数は減少している．図 10 は，3.3 で示した最適化（NUMA Control，First Touch Data Placement，連続データアクセスのための再データ配置）を適用した後の，T2K（東大）を使用した場合の，各プログラミングモデルの各色における計算性能である．

図 9 に示したように，色数が増えると反復回数が減少しているにもかかわらず，図 10 (a) に示すように，各並列プログラミングモデルにおいて，色数＝2 の場合（CM-RCM(2)）の計算時間（MGCG ソルバーの計算時間）が最も短い．反復あたりの計算時間についても，図 10 (b) に示したように，CM-RCM(2)が他と比べて小さく，性能が高いことがわかる．これは，本研究で対象としているような単純な差分格子形状の場合，CM-RCM 法においては，色数が少ないほどキャッシュを有効利用できるためである．図 11 (a)，(b) は 64 セルを有する二次元差分格子における例である．

- 図 11 (a) CM-RCM(2) (#1-#32 番のセルが第 1 色，#33-#64 番のセルが第 2 色に属している)
- 図 11 (b) RCM (15 色)

ILU プロセスにおける前進後退代入（forward/backward substitutions，FBS）を考慮する場合，例えば CM-RCM(2) の#29，#30，#31 番セルの非対角成分の番号（#59～#64）は連続であり，対角成分と非対角成分は独立したキャッシュライン上に乗っている（図 11 (a)）．それに対して RCM の場合は対応する対角成分，非対角成分の番号は連続しており（#55～#63），同一のキャッシュライン上に並ぶ可能性がある（図 11 (b)）．従って，RCM の場合，前進後退代入時に同じキャッシュライン上の変数値が変わり，キャッシュからメモリに書き戻され，効率が低下する可能性がある．

45	10	39	5	35	2	33	1
17	46	11	40	6	36	3	34
53	18	47	12	41	7	37	4
24	54	19	48	13	42	8	38
59	25	55	20	49	14	43	9
29	60	26	56	21	50	15	44
63	30	61	27	57	22	51	16
32	64	31	62	28	58	23	52

(a) CM-RCM(2)

29	22	16	11	7	4	2	1
37	30	23	17	12	8	5	3
44	38	31	24	18	13	9	6
50	45	39	32	25	19	14	10
55	51	46	40	33	26	20	15
59	56	52	47	41	34	27	21
62	60	57	53	48	42	35	28
64	63	61	58	54	49	43	36

(b) RCM (15 colors)

図 11　64 セルを有する二次元差分格子の例

図 12 は T2K（東大）において，3.3 で述べた最適化の効果を各並列プログラミングモデルについて CM-RCM(2)の場合について比較したものである．実行時に NUMA control を適用することで，特に OpenMP/MPI ハイブリッド並列プログラミングモデルは 2 倍以上の高速化が可能である．更に，First Touch Data Placement，連続データアクセスのための再データ配置(図 5)を適用することにより，

Flat MPI と OpenMP/MPI ハイブリッド並列プログラミングの性能はほぼ同等となる.

HB 4 4 では，もともとデータが各ソケットのローカルメモリに配置されているため，HB8×2，HB16×1 と比較して性能が高い．収束までの反復回数は同じ 64 回であるが，計算時間はそれぞれ，21.8sec.（HB 4 4），22.6sec.（HB 8 2），23.2sec.（HB 16 1）である．したがって，HB 4 4 では，First Touch Data Placement と再データ配置の効果もわずかである.

図 13 は，図 10（b）に相当するもので，MGCG 法の計算性能と CM-RCM の色数の関係について，1 反復あたりの計算時間を T2K（東大）と Cray-XT4 を比較したものである.

Cray-XT4 の実効性能は T2K（東大）よりも 50%程度高い．表 2 に示すように，STREAM ベンチマーク〔13〕で測定したメモリの性能は 2 倍近く Cray-XT4 が高い.

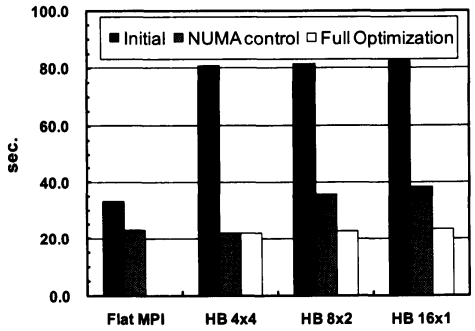


図 12　MGCG 法の計算性能と最適化の効果（MGCG ソルバー計算時間）（CM-RCM(2)，16,777,216 セル，T2K（東大），64 コア）（不均質多孔質媒体中の三次元地下水流れ）（■Initial：初期状態，■NUMA control：実行時の最適な NUMA control の適用，□Full Optimization：NUMA Control, First Touch Data Placement, 連続データアクセスのための再データ配置を全て適用した場合）

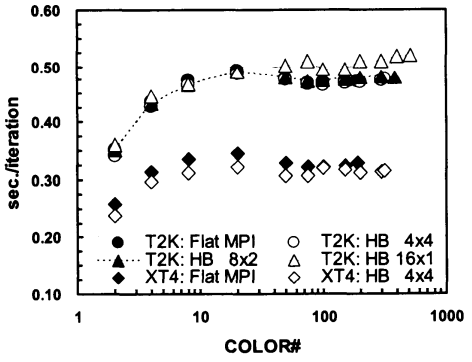


図 13　MGCG 法の計算性能と CM-RCM 法の色数の関係（MGCG ソルバー1 反復あたり計算時間）（16,777,216 セル，64 コア）（不均質多孔質媒体中の三次元地下水流れ）（3.3 に示した最適化（NUMA Control, First Touch Data Placement, 連続データアクセスのための再データ配置（図 8）適用後））

表 2　T2K（東大），Cray-XT4 の性能概要

	T2K（東大）	Cray-XT4
STREAM: Triadd [12] (GB/sec/core)	1.23	2.35
GeoFEM Benchmark [13] (MFLOPS/core)	292	512

4.2 大規模問題（Weak Scaling）

T2K（東大），Cray-XT4 の 16～8,192 コアを使用して，大規模問題に対する性能と安定性を評価した．まず，コアあたりの問題規模を固定した Weak Scaling によって評価を実施した．コア当たり問題サイズ（セル数）は 262,144 ($=64^3$) であり，最大問題規模は 2,147,483,648 セルである．問題設定は 4.1 と同じであり，図 12 に示した最適化されたソルバーを使用し，CM-RCM(2)を適用した.

ここでは，最も粗い格子レベル（図 5 における Level=k）での解法（Coarse Grid Solver）について 3 つの手法を比較した．3．でも述べたように，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，最も粗い格子レベル（図 5 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＝格子数となる．以下の 3 種類の手法を比較した：

- **C0:** 各領域 1 セルになった状態で 1 コアに集め，IC(0) スムージングを 2 回施す（4.1 の手法と同じ）
- **C1:** IC(0)スムージングを収束（$\varepsilon=10^{-12}$）まで繰返す
- **C2:** マルチグリッド（V-cycle）を適用し，収束（$\varepsilon=10^{-12}$）まで繰返す

従って，C1，C2 では，C0 と比較して 1 コアでの計算量が大きくなるものと考えられる.

図 14 は，16 コア～8,192 コアを利用した場合の MGCG 法（C0）の T2K（東大）における計算性能である．図 14（a）は収束までの反復回数である．完全にスケーラブルな場合は Weak Scaling によって，反復回数は変化せず，計算時間も変化しないが，本研究で扱っているような悪条件問題では，問題規模が大きくなるに従って，反復回数が増加している．また，領域間通信の影響のため，計算時間もコア数とともに増加し，コア数が 256（問題規模としては約 10^8 セル）を超えると，特に Flat MPI の反復回数の増加が顕著になる．Flat MPI では 16 コアと 8,192 コアを比較すると，反復回数で約 3 倍，計算時間で約 4 倍となっている.

それと比較すると，OpenMP/MPI ハイブリッド並列プログラミングモデルの場合の増加は Flat MPI ほど顕著では無い．図 14（a）に見られるように，HB 16 1 は他の並列プログラミングモデルと比較して，反復回数の増加は低く抑えられている．HB 16 1 の場合では，16 コアと 8,192 コアを比較すると，反復回数は 57 回から 84 回に増加し，計

算時間は約 2 倍強に増加している．

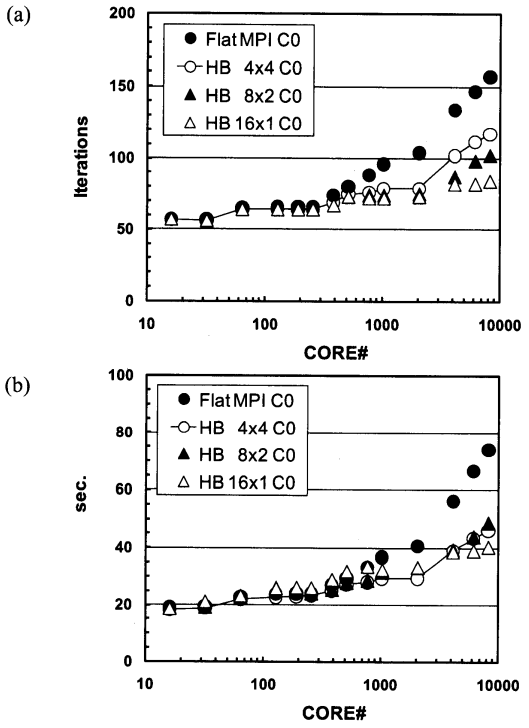


図 14　MGCG 法の計算性能（C0）（(a) 反復回数，(b) MGCG ソルバー計算時間）（最適化されたソルバーを適用，CM-RCM(2)，16～8,192 コア，T2K（東大））（Weak Scaling，262,144 セル／コア，最大問題規模：2,147,483,648 セル）（不均質多孔質媒体中の三次元地下水流れ）

図 15 は Flat MPI について，C0, C1, C2 の 3 種類の Coarse Grid Solver の効果を比較したものである．256 コアまでは 3 者の違いは全く無いが，256 コアを超えても C1，C2 の反復回数はほとんど増加していないことがわかる．C0 では一番粗いレベルの格子で IC(0)によるスムージングを 2 回施しているだけであるが，コア数が増加するとそれだけ一番粗いレベルの格子数が増加するため，2 回のスムージングでは緩和が不十分であると考えられる．

計算時間については，2,048 コアまでは C1 と C2 の差はほとんど無いが，2,048 コアを超えると C1 の計算時間が急激に増加する．C1 では IC(0)スムージングを収束（$\varepsilon=10^{-12}$）まで繰り返しているため，コア数が増加するとそれだけ Coarse Grid Solver における問題規模，収束までの反復回数が増加するためと考えられる．図 16 は図 15 の各ケースにおける Coarse Grid Solver の計算時間の比較である．各ケースともコア数，すなわち問題規模が増加するとともに Coarse Grid Solver すなわち 1 コアで計算する時間は増加しており，この傾向は特に C1 の場合に顕著である．C2 はこれに比べると約 10 分の 1 程度であるが，それでも C0 の場合と比較すると約 1 オーダー大きい．C2 の場合，計算時間全体に占める割合は 8,192 コアの場合約 6%程度である．

しかしながら，C2 を C0 と比較すると，反復回数が減少しているため，計算時間は 8,192 コアで半分以下となっている．

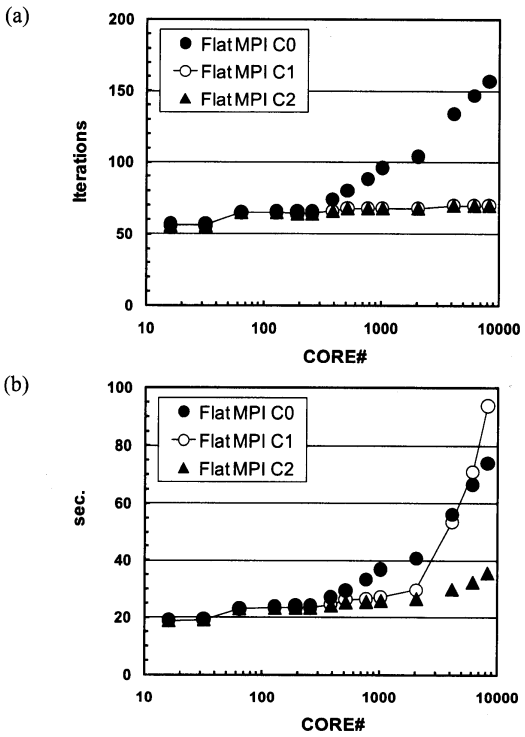


図 15　MGCG 法の計算性能（Coarse Grid Solver の効果：C0，C1，C2）（(a) 反復回数，(b) MGCG ソルバー計算時間）（最適化されたソルバーを適用，CM-RCM(2)，16～8,192 コア，T2K（東大），Flat MPI）（Weak Scaling，262,144 セル／コア，最大問題規模：2,147,483,648 セル）（不均質多孔質媒体中の三次元地下水流れ）

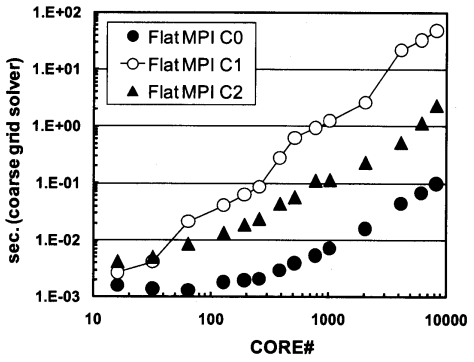


図 16　MGCG 法における Coarse Grid Solver の計算時間（C0, C1, C2）（最適化されたソルバーを適用，CM-RCM(2)，16～8,192 コア，T2K（東大），Flat MPI）（Weak Scaling，262,144 セル／コア，最大問題規模：2,147,483,648 セル）（不均質多孔質媒体中の三次元地下水流れ）

図 17 は Coarse Grid Solver に C2 を適用した場合の各並列プログラミングモデルの計算性能（反復回数，MGCG ソルバーの計算時間）である．図 14 と比較すると，コア数増加による反復回数の増加は抑制され，よりスケーラブ

ルになっていることがわかる．

8,192コアにおける収束までの反復回数とMGCGソルバーの計算時間は以下の通りであり，並列プログラミングモデルによる差異は少なくなっている：

- **Flat MPI**: 70回，35.7sec.
- **HB 4×4**: 71回，28.4sec.
- **HB 8×2**: 72回，32.8sec.
- **HB 16×1**: 72回，34.4sec.

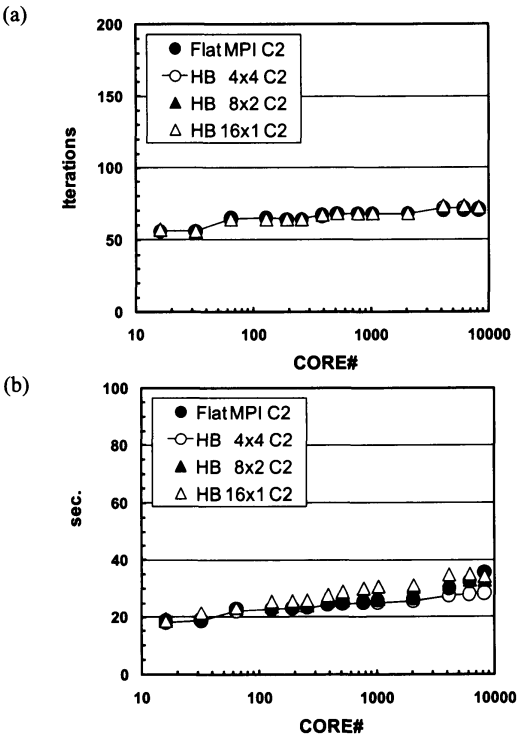


図 17 MGCG 法の計算性能（C2）（(a) 反復回数，(b) MGCG ソルバー計算時間）（最適化されたソルバーを適用，CM-RCM(2)，16～8,192 コア，T2K（東大））（Weak Scaling，262,144 セル／コア，最大問題規模：2,147,483,648 セル）（不均質多孔質媒体中の三次元地下水流れ）

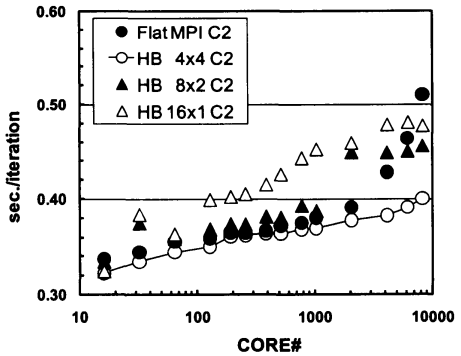


図 18 MGCG 法の計算性能（C2）（MGCG ソルバー1 反復あたり計算時間）（最適化されたソルバーを適用，CM-RCM(2)，16～8,192 コア，T2K（東大））（Weak Scaling，262,144 セル／コア，最大問題規模：2,147,483,648 セル）（不均質多孔質媒体中の三次元地下水流れ）

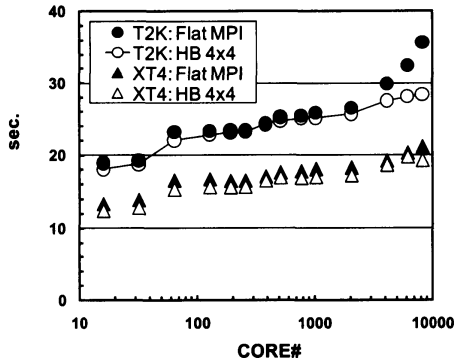


図 19 MGCG 法の計算性能（C2）（MGCG ソルバー計算時間）（最適化されたソルバーを適用，CM-RCM(2)，16～8,192 コア，T2K（東大），Cray-XT4）（Weak Scaling，262,144 セル／コア，最大問題規模：2,147,483,648 セル）（不均質多孔質媒体中の三次元地下水流れ）

また，図 18 は 1 反復あたりの計算時間である．収束までの計算時間，計算効率（1 反復あたり計算時間）ともに HB 4×4 が最も性能が高いことがわかる．

図 19 は C2 に関して T2K（東大）と Cray-XT4 の計算性能を比較したものである．図 13 と同じ傾向であり，Cray-XT4 は T2K（東大）と比較して 40%～50%程度高速である．

4.3 大規模問題（Strong Scaling）

続いて，全体の問題規模を 33,554,432 セル（=512×256×256）に固定し，コア数を 16 から 1,024 まで変化させた Strong Scaling による評価を実施した．問題設定は 4.1 と同じであり，図 12 に示した最適化されたソルバーを使用し，CM-RCM(2)を適用した．また Coarse Grid Solver としては C2 を使用した．図 20 は T2K（東大）における Strong Scaling の結果である．Flat MPI，16 コアの場合を 100%として並列化効率を求めている．1,024 コアにおける効率は HB 4×4 の場合が，約 74%で最も高い．

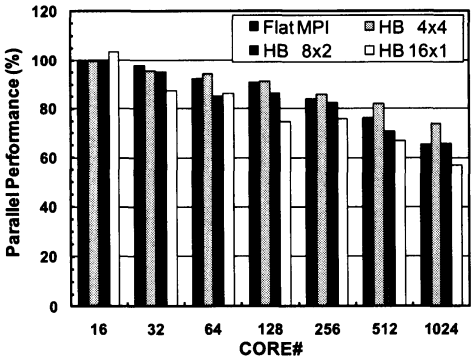


図 20 MGCG 法の計算性能（C2）（MGCG ソルバー計算時間，Flat MPI・16 コアの場合を 100%とする）（最適化されたソルバーを適用，CM-RCM(2)，16～1,024 コア，T2K（東大））（Strong Scaling，問題規模：33,554,432 セル（=512×256×256））（不均質多孔質媒体中の三次元地下水流れ）

図21はFlat MPI, HB 4×4の場合について, T2K(東大), Cray-XT4のStrong Scalingの性能をFlat MPI, 16コアの場合を100%として比較したものである. Flat MPI, HB 4×4の性能の相対的関係はT2K（東大）とCray-XT4で同様であるが，コア数を増加した場合の性能低下はCray-XT4においてより顕著である. これは，図13，図19等で示したように，コアあたりの実効計算性能においてCray-XT4がT2K（東大）を40%～50%程度上回っているため，相対的に通信による影響が大きくなるためと考えられる.

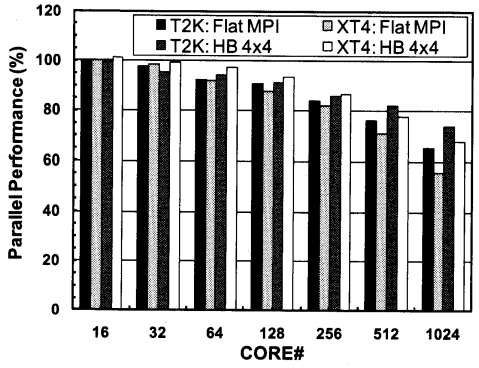


図21　MGCG法の計算性能（C2）（MGCGソルバー計算時間，Flat MPI・16コアの場合を100%とする）（最適化されたソルバーを適用，CM-RCM(2)，16～1,024コア，T2K（東大），Cray-XT4）（Strong Scaling，問題規模：33,554,432セル（=512×256×256））（不均質多孔質媒体中の三次元地下水流れ）

5. 関連研究

多重格子法は大規模シミュレーションにおいて重要な手法であるが，OpenMP/MPI Hybridを並列多重格子法に適用した例はまだ少ない. 本研究の他には，Baker等の研究例〔14〕があるのみである.〔14〕ではHypre Library（BoomerAMG）〔15〕を対象として，T2K（東大）と同じノード構成（AMD quad-core Opteron（2.3GHz）4ソケット／ノード，4コア／ソケット）を持つマルチコア，マルチソケットクラスタ（Hera（Lawrence Livermore National Laboratory, USA）〔16〕）において216ノード，3,456コアを使用して本研究と同様にFlat MPI，様々なハイブリッド並列プログラミングモデルの比較を実施し，HB 4×4が最も高い性能を示している. またMultiCore SUPport library（MCSup）〔14〕というライブラリにより，本研究におけるFirst Touch Data Placement，Sequential Reordering等と同様な機能を自動的に実施するフレームワークを提供している.

6. まとめ

OpenMP/MPIハイブリッド並列プログラミングモデルを，並列多重格子前処理付き反復法を使用した，三次元有限体積法に基づく不均質多孔質媒体中における地下水流れ問題シミュレーションに適用した. 多重格子法の緩和演算子としては，IC(0)を適用した. 開発したプログラムの性能と安定性をT2Kオープンスパコン（東大），Cray-XT4の8,192コアまでを使用して評価した.

First Touch Data Placement, 連続メモリアクセスのためのデータ再配置, 適切なNUMA controlの組み合わせにより，OpenMP/MPIハイブリッド並列プログラミングモデルがFlat MPIと同等かそれを上回る性能を発揮することがわかった. またコア数が増加した場合には，適切なCoarse Grid Solverの選択が重要であることがわかった.

OpenMP/MPIハイブリッド並列プログラミングモデルの中では特に各ソケットに1つのMPIプロセスを割り当てるHB 4×4が，最も高い性能を示している. 並列多重格子法では，粗いレベルでの通信のオーバーヘッドが大きくなるため，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Flat MPIと比較して有利である. また，各ソケットに1つのMPIプロセスを割り当てる手法（本研究ではHB 4×4）は，ローカルなデータがソケットのローカルメモリ上にあることが保証されているため，他のハイブリッド並列プログラミングモデルと比較してメモリアクセス上有利であると考えられる.

今後の研究課題としては，特に粗い格子レベルにおける演算の並列性能の向上と安定化があげられる. 本研究でも明らかになったように，Coarse Grid Solverは計算の効率に大きな影響を及ぼす. 粗い格子レベルでのMPIの立ち上がりのLatencyによるオーバーヘッドの効果は，MPIプロセス数が多い場合に特に顕著となる. 粗い格子レベルにおいては，段階的に複数のプロセスを統合してMPIプロセス数を減らすような工夫が必要である.

また，現状ではハイブリッド並列プログラミングモデルにおいて，CM-RCMリオーダリングのプロセスは並列化されていない. 今後のマルチコア化，メニーコア化に備えて，この部分の並列化も重要な課題である.

本研究の成果は，昨今科学技術計算で使用されるようになっているGPUに関しても適用可能であり，今後有効に活用して行きたい.

参考文献

[1] Nakajima, K.: Flat MPI vs. Hybrid: Evaluation of Parallel Programming Models for Preconditioned Iterative Solvers on "T2K Open Supercomputer", IEEE Proceedings of the 38th International Conference on Parallel Processing (ICPP-09), pp.73-80 (2009)

[2] Information Technology Center, The University of Tokyo: http://www.cc.u-tokyo.ac.jp/

[3] 中島研吾, 不均質場におけるマルチレベル解法, ハイパフォーマンスコンピューティングと計算科学シンポジウム HPCS2006 論文集, pp.95-102 (2006)

[4] NERSC, Lawrence Berkeley National Laboratory:

http://www.nersc.gov/

[5] The T2K Open Supercomputer Alliance: http://www.open-supercomputer.org/

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

[7] Tottemberg, U., Oosterlee, C. and Schuller, A.: Multigrid, Academic Press (2001)

[8] Nakajima, K.: Parallel Multilevel Iterative Linear Solvers with Unstructured Adaptive Grids for Simulations in Earth Science, Concurrency and Computation: Practice and Experience 14-6/7, pp.484-498 (2002)

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

[10] 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)

[11] Mattson, T.G, Sanders, B.A., Massingill, B.L.: Patterns for Parallel Programming, Software Patterns Series (SPS), Addison-Wesley (2005)

[12] 中島研吾，片桐孝洋，マルチコアプロセッサにおけるリオーダリング付き非構造格子向け前処理付反復法の性能，情報処理学会研究報告（HPC-120-6）（2009）

[13] STREAM (Sustainable Memory Bandwidth in High Performance Computers): http://www.cs.virginia.edu/stream/

[14] Allison Baker, Martin Schultz, Ulrike, Yang, On the Performance of an Algebraic Multigrid Solver on Multicore Clusters, 9th International Meeting High Performance Computing for Computational Science (VECPAR 2010) (2010)

[15] http://acts.nersc.gov/hypre/

[16] https://computing.llnl.gov/?set=resources&page=OCF_resources#hera