

戦略的創造研究推進事業 SPPEXA(CREST)  
研究領域「ポストペタスケール高性能計算に資する  
システムソフトウェア技術の創出」  
研究課題「自動チューニング機構を有するアプリケーション開発・実行環境」

## 研究終了報告書

研究期間 平成27年12月～平成30年3月

研究代表者：中島 研吾  
(東京大学情報基盤センター、教授)

## § 1 研究実施の概要

### (1) 実施概要

平成 23 年度～27 年度に実施された「自動チューニング機構を有するアプリケーション開発・実行環境『ppOpen-HPC』」の研究開発において得られた知見、ソフトウェア資産を元に「ポスト京」等メニーコアアーキテクチャ向けのアプリケーション開発フレームワーク「pK-Open-HPC」を開発した。pK-Open-HPC は、ppOpen-HPC の機能を拡張した以下の 3 コンポーネントから構成される:

- ・ pK-Open-FVM: ブロック AMR 型有限体積法に基づくアプリケーション開発フレームワーク
- ・ pK-Open-SOL: 高性能な疎行列計算向け前処理付き反復法ソルバ
- ・ pK-Open-AT: パラメータ自動チューニング機構

pK-Open-FVM では、これまでの成果を元に様々な分野のアプリケーションへの応用を図り、プラズマ蒸着装置開発へ向けたシミュレーション、Vlasov-Poisson 系シミュレーションなどを実施する一方で、Block 型 AMR フレームワークを開発し、エージェントベースモデルに基づく、粒子系シュガースケープ (PSS) モデルシミュレーションを実施し、従来のプラズマ物理以外の分野へも積極的に適用を図り、これらの実シミュレーション結果を考慮して、フレームワークの改良を実施した。

pK-Open-SOL ではドイツ DFG の SPPEXA-II の一環として実施されている ESSEX-II プロジェクト (Equipping Sparse Solvers for Exascale, Leading PI: Professor Gerhard Wellein (FAU Erlangen)) と連携して、量子力学アプリケーションから導出される大規模な固有値問題の並列前処理付き反復法による解法の研究を実施した。これらの問題は一般に対角成分の絶対値が小さく、従来の反復法による求解は非常に困難であることが知られているが、本研究では、ブロック化、対角シフトの導入による安定で効率的な前処理手法を提案し、従来求解不可能であった問題も解けるようになった。これらの成果は ESSEX, ESSEX-II の成果物である GHOST および PHIST ライブラリへの統合を進めている。また、分散並列環境下における、効率的で安定な前処理手法の実現のために、階層的な多色順序付け法を提案、実装し、予備的な評価についてまとめた論文が IEEE HPCC 2017 で採択された。

ppOpen-AT (ppOpen-HPC の自動チューニング機構) では、ディレクティブベースの AT (Automatic Tuning, 自動チューニング) 向けの専用言語を開発し、ステンシル計算、密行列演算、疎行列演算などで大きな成果を上げてきた。pK-Open-AT ではステンシル計算、疎行列向け前処理付き反復法等について、ESSEX チームと協力して性能評価モデルを構築し、より効果的な AT フレームワークの開発を目指している。本研究では、有限差分法アプリケーション (ppOpen-APPL/FDM) の主演算部分において、従来の実装に加え、①Byte per FLOPS が小さくなる実装方式、および、②更に①に加えて IF 文がない実装によりコンパイラ最適化を推進する実装方式、を開発した。また、この 3 種の実装方式について最適な実装を自動チューニング (AT) により選択する階層型 AT 方式を開発し、様々な並列計算機システム上で検証した。

大規模な並列計算機を使用する場合、ノード数の増加によって通信のオーバーヘッドは増加する傾向にある。並列計算において通信は必須のプロセスであるが、通信をできる限り効率的に実施し、削減することはエクサフロップス級システムにおいて重要である。本研究では、クリロフ部分空間法に焦点を当て、パイプライン法、動的ループスケジューリングに関するフィージビリティスタディを様々な計算機システムを使用して実施した。特に動的ループスケジューリングは、2016 年 12 月に全系運用を開始した Oakforest-PACS システム (Xeon Phi/Knights Landing 8,208 ノード、ピーク性能 25PFLOPS) では大きな効果を発揮し、並列有限要素法アプリケーションに適用した場合に 25% 近い速度向上率を得ることができた。

ESSEX-II プロジェクトでは、エクサフロップス級システムにおいて Check-Point に基づく Fault Tolerance を実現するライブラリとして、CRAFT (A library for application-level

Checkpoint/Restart & Automatic Fault Tolerance) を開発している。ppOpen-HPC/pK-Open-HPC の動的負荷分散機能を使用して、超並列環境において、有限要素法等のシミュレーションを実行中に計算ノードにハードエラーが生じた場合に、いわゆるスペアノードを使用することなく、残りの計算資源を使用してシミュレーションを継続するフレームワークを開発中である。当初の研究計画にはなかった項目であるが、2017年6月から本格的に共同で開発を進めている。

## (2) 顕著な成果

### <優れた基礎研究としての成果>

#### 1. pK-Open-SOL における前処理付き反復法

##### 概要:

pK-Open-SOL では量子力学の分野から導出されるような悪条件な連立一次方程式を想定した安定で効率的なソルバを開発した。本ソルバの適用によりこれまで求解困難とされてきた問題が解けることを検証した。本ソルバを PHIST ライブラリの一部として公開し、ヘテロジニアス環境を含む様々なシステムで高い性能を実現することを確認した。本成果は今後の科学技術の発展に貢献すると期待される。

#### 2. pK-Open-SOL における並列多色順序付け手法

##### 概要:

pK-Open-SOL で新たに提案した多色順序付け用の階層的な並列化手法は、既存の多色順序付けアルゴリズムを並列化可能な汎用的な手法である。多色順序付けの結果は ILU 前処理の収束性に強く影響するため、汎用的である点は重要である。また、多色順序付けの並列化と ILU 前処理の関係性に関する研究報告は非常に少なく、エクサスケールを想定した研究として先導的である。

### <科学技術イノベーションに大きく寄与する成果>

#### 1. pK-Open-FVM の開発

##### 概要:

有限体積法や差分法で離散化されているアプリケーションへ適合格子細分化法 (AMR) 法を導入するための Block ベース AMR フレームワークである pK-Open-FVM の開発を行った。本フレームワークは、ひとまとめにした計算格子群をブロックと定義し、そのブロック自体が生成・消滅することで AMR 機能を実現するため、移植性に優れる点が最大の特徴であり、従来計算が難しかったプラズマ蒸着装置開発へ向けた解析や、高精度エージェントベースシミュレーション等への適用が進められた。

#### 2. pK-Open-AT の開発

##### 概要:

有限差分法アプリケーション (ppOpen-APPL/FDM) の主演算部分において、従来のループ分割/ループ融合のループ変換に加え、Byte per FLOPS が小さくなる実装方式のコード選択方式を実装した。また、最新鋭メニーコア CPU である Xeon Phi (KNL) のメモリとスレッド割り当てモード 4 種の性能評価を世界に先駆けて行い、同環境における AT 方式の有効性を示した。一方、開発した AT 方式が核融合コードにおいても適用可能なことを初めて示し、計算科学分野の技術イノベーションに貢献した。

#### 3. 並列クリロフ部分空間法の通信最適化

##### 概要:

大規模な並列計算機を使用する場合、ノード数の増加によって通信のオーバーヘッドは増加する傾向にある。並列計算において通信は必須のプロセスであるが、通信をできる限り効率的に実施し、削減することはエクサフロップス級システムにおいて重要である。本研究では、ク

リロフ部分空間法に焦点を当て、パイプライン法、動的ループスケジューリングに関するフィージビリティスタディを様々な計算機システムを使用して実施した。

## § 2 研究実施体制

(1) 研究チームの体制について

### ① 「中島」グループ

研究参加者

氏名	所属	役職	参加時期
中島 研吾	東京大学情報基盤センター	教授	H27.12～
片桐 孝洋	名古屋大学情報基盤センター	教授	H27.12～
伊田 明弘	東京大学情報基盤センター	特任准教授	H27.12～
岩下 武史	北海道大学情報基盤センター	教授	H27.12～
松本 正晴	東京大学情報理工学系研究科	特任講師	H27.12～ (SPPEXA 雇用は H28.4-H.28.11)
大島 聡史	九州大学情報基盤研究開発センター	助教	H27.12～
佐藤 正樹	東京大学大気海洋研究所	教授	H27.12～H20.3
古村 孝志	東京大学地震研究所	教授	H27.12～
奥田 洋司	東京大学大学院・新領域創成科学研究科人間環境学専攻	教授	H27.12～
阪口 秀	海洋研究開発機構	分野長	H27.12～
荒川 隆	高度情報科学技術研究機構	主任研究員	H27.12～
河合 直聡	東京大学情報基盤センター	特任助教	H28.4～
星野 哲也	東京大学情報基盤センター	助教	H28.4～

研究項目

- ・ pK-Open-FVM(ブロック AMR 型有限体積法に基づくアプリケーション開発フレームワーク)
- ・ pK-Open-SOL(高性能な疎行列計算向け前処理付き反復法ソルバ)
- ・ pK-Open-AT(性能評価モデルと連携したパラメータ自動チューニング機構)

## § 3 研究実施内容及び成果

### 3.1 pK-Open-FVM の開発(東京大学 中島グループ)

(1) 研究手法の概要

有限体積法や差分法で離散化されているアプリケーションへ適合格子細分化法(Adaptive Mesh Refinement, AMR)法を導入するための自己相似型 Block ベース AMR フレームワークである pK-Open-FVM の開発を進めており、公開を予定している。計算格子の1つ1つが生成・消滅する Cell ベースの AMR シミュレーションとは異なり、自己相似型 Block ベース AMR シミュレーションでは、ある程度ひとまとめにした計算格子群をブロックと定義し、そのブロックが生成・消滅することで、AMR 機能を実現する(図 1 参照)。これにより、有限体積法や差分法などでよく用いられる  $i, j, k$  などのインデックスで整理される整合配列データ構

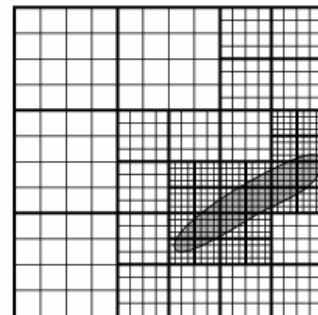


図 1 自己相似型 Block-AMR 例  
(1ブロック当たり4×4格子、計3階層の2次元例)

造で計算されるkernel部分をそのままフレームワークのkernelとして使用することができ、移植性に優れ簡単にAMRシミュレーションを行える、という点がBlock-AMRフレームワークの最大の特徴である。このBlock-AMRフレームワークでは、計算領域は複数に分割され、MPIを使用した並列計算のための複数のプロセスに割り当てられる。MPIプロセスに対応するそれぞれの分割された計算領域は、それぞれが固定数のグリッドを有する複数のブロック構造領域から構成され、計算領域の特定の領域で高解像度が要求される場合、細分化された格子を持つブロック構造領域が子ブロックとして作成される〔原著 1,10〕。

pK-Open-FVM は ppOpen-HPC の一部として、<http://ppopenhpc.cc.u-tokyo.ac.jp/ppopenhpc/>より公開済みである。以下では、pK-Open-FVMを用いたシミュレーション例として、1) 格子細分化を行わない一様(均一メッシュ)次元と、格子細分化を行うAMR次元を組み合わせるVlasov-Poissonシミュレーション、2) プラズマ蒸着装置開発のためのプラズマ粒子を導入したAMR粒子シミュレーション、そして、プラズマ分野以外への応用として3) 粒子系シュガースケープモデルへの応用についてそれぞれ述べる。

## (2) Vlasov-Poisson 系シミュレーションへの応用

AMR 計算による効率化は超大規模計算を行う上で大きな魅力であり、特に 4 次元以上の高次元計算が一つの対象として考えられる。プラズマ物理シミュレーション分野において代表的な高次元計算の一つに Vlasov 方程式の数値解法がある。Vlasov 方程式は

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \frac{\partial f}{\partial \mathbf{x}} + \frac{\mathbf{F}}{m} \cdot \frac{\partial f}{\partial \mathbf{v}} = 0$$

の形で表され、Boltzmann 方程式の右辺に現れる衝突項を省略した式である。ここで、 $f = f(\mathbf{x}, \mathbf{v}, t)$  は速度分布関数、 $\mathbf{x}, \mathbf{v}$  はそれぞれ位置、速度、 $m$  は粒子の質量、 $\mathbf{F}$  は外力を表している。これは実空間ベースで考えると、位置座標3次元+速度座標3次元の計6次元計算となり、それらが上式第二項の速度と空間偏微分の積、第三項の外力と速度偏微分の積によって相互に影響を与えるが、この式について6次元空間を定義して計算を行うことは、計算規模の面から非常に多くのメモリを必要とするため、あまり現実的ではない。また、例え6次元計算に1次元方向2分割のAMR機能を導入したとして、6次元空間では、1回の細分化に64ブロックも生成されることとなり、このような爆発的に増減するブロックをメモリ上で動的に管理し、その接続関係を構築し続けるということは非常に困難である。

そこで、格子細分化を行わない一様(均一メッシュ)次元と、格子細分化を行うAMR次元の組み合わせを可能にするBlock-AMRフレームワークの拡張を行い、空間1次元・速度1次元の2次元Vlasov-Poisson系に対し、本フレームワークを適用する。Vlasov-Poissonシミュレーションでは、外力としての電場を計算するために空間方向にPoisson方程式を解く必要があることから、空間方向を一様次元と置き、高速Fourier変換によるPoissonソルバを適用するとともに、速度方向をAMR次元とする。図2に、代表的な条件で計算を行った際の時間  $t = 0, 128, 176\text{s}$  での分布関数のコンター図を示す。計算したモデルはビーム不安定性と呼ばれているもので、2つの速度成分を持つ電子ビームを入力し時間を経過させるとそのビーム間で渦を発生させつつ、徐々に一つの電子ビームへと混ざり合っていくというモデルである。本計算例では拡張フレームワークの実行テストとして、ビーム間で渦ができるまでの状態を計算し、細分化条件としてAMR次元方向、すなわち速度空間方向に勾配がある場合に階層を生成するよう設定し、渦部分を高解像度に計算した。図2の右下に渦部分の一部波形の拡大図を示しており、波形の勾配が大きい部分に最大解像度であるLv.3ブロックが生成される。また計算時間は一様格子を用いる場合と比較して、約30%程度低減することができた。

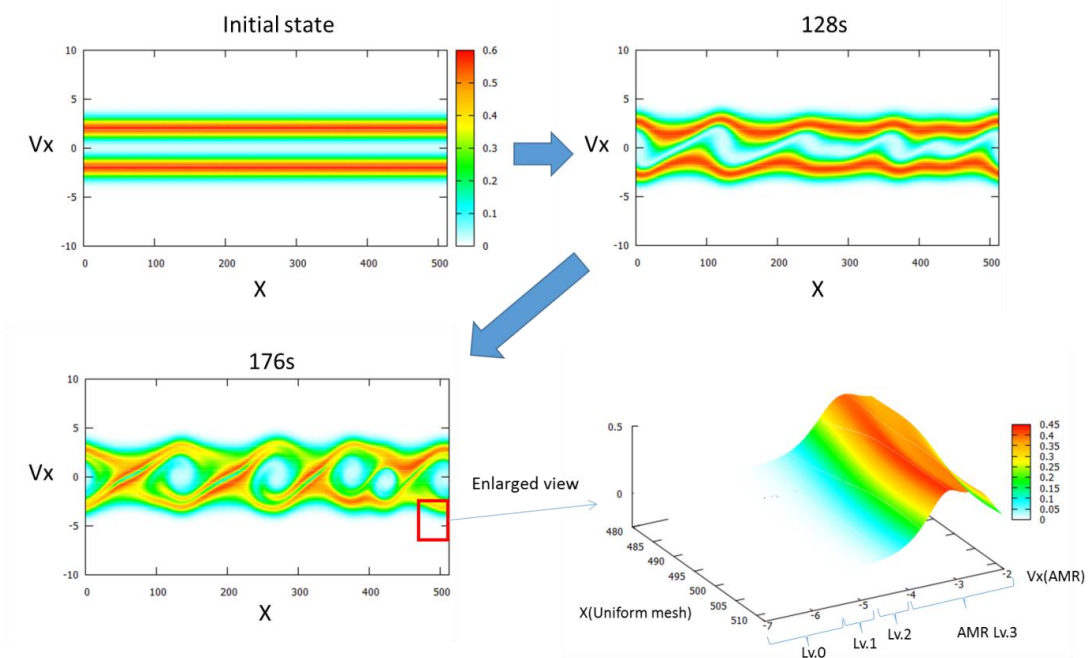


図 2 1D-1D Vlasov-Poisson 系シミュレーションへの AMR の導入

(3) プラズマ蒸着装置開発へ向けた AMR 粒子シミュレーション

企業との共同開発として、pK-Open-FVM を用いて、プラズマコーティング装置の一種である反応性プラズマ蒸着 (Reactive Plasma Deposition, RPD) 成膜装置 (図3参照) 開発へ向けた、プラズマ粒子/流体ハイブリッドシミュレーションコードの開発を進めている。

このシミュレーションでは、格子点上に定義される電磁場と、メッシュフリーな粒子として取り扱われるプラズマ中のイオン・電子を連立して計算することで、電磁場とプラズマ粒子の相互作用を解き進めることができる。このようなプラズマ粒子を取り扱うシミュレーションでは、粒子数に依存

する統計誤差の影響を低減するために、1 格子当たりの粒子数を数百個程度以上用意する必要があるが、従来提案の一樣格子を用いるシミュレーションでは、計算領域内でプラズマ密度の高低差が大きくなる場合、プラズマ密度が低い領域で粒子数が少なくなり、そこでの統計誤差が大きくなるため、これを是正するために更に多くの粒子を必要とする、という問題が生じ計算時間が増大する。そこで、AMR フレームワークを利用して、格子間隔をプラズマ密度と対応付けることで、低密度領域では格子間隔を広く、高密度領域では狭くし、全体の粒子数を増やすことなく1格子当たりの粒子数を適切な数に調整できるため、結果として、全体の粒子数を低減でき計算時間を早めることが可能となる (図4、5参照)。従来提案のシミュレーション手法ではできなかった、材料源として酸化インジウムスズを用いた計算では、実験における膜厚分布のコイル電流依存性を定性的に再現でき、現在はこのシミュレーションをベースに新たな装置の開発に着手している。

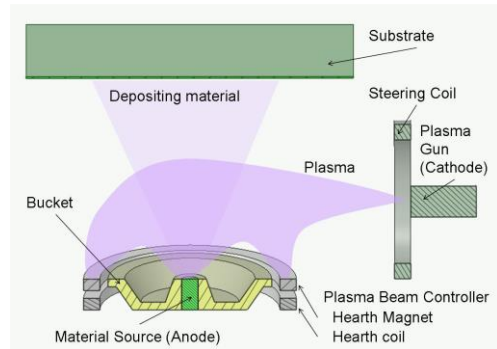


図3 RPD 装置概略図

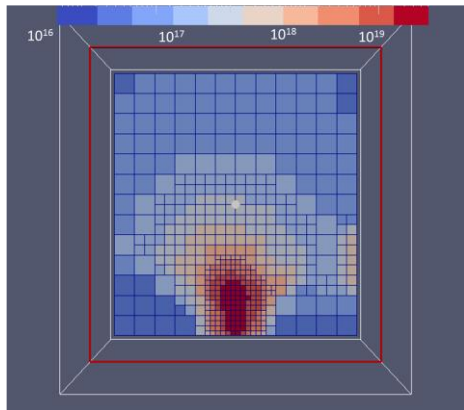


図4 イオン密度分布と計算格子分布

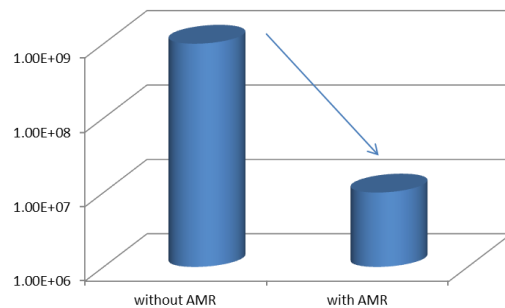


図5 AMR 機能の有無による計算領域内粒子数の変化

#### (4) 粒子系シュガースケープ (PSS) モデルへの応用

AMR フレームワークのプラズマシミュレーション分野以外への応用として、ある地域の住民を代表するエージェントを多数用いて、人工社会シミュレーションのために提案されている粒子系シュガースケープ (PSS) モデルを採用し、テストシミュレーションを行った。シミュレーション領域の各格子には糖度が割り当てられており(図6参照)、その初期条件として砂糖の2つのピークを配置するとともに、(粒子である)エージェントをランダムに配分する。PSS モデルでは、エージェント粒子は格子点に依存せずにシミュレーション領域内を移動する。粒子の位置、及び速度の時間変化は以下の運動方程式を解き進めることによって得られる。

$$\begin{cases} \frac{\partial \mathbf{x}}{\partial t} = \mathbf{v} \\ \frac{\partial \mathbf{v}}{\partial t} = m\mathbf{F}_s - n\mathbf{F}_d \end{cases}$$

ただし、 $\mathbf{F}_s$ は糖度密度場の勾配による力、 $\mathbf{F}_d$ はエージェント密度場の勾配による力をそれぞれ示している。つまり、これらの式より、エージェント粒子は糖度の高い方へ、かつ、エージェント密度が低い方へ進むように力を受ける。

シミュレーション結果は予想通り、エージェントは2つの糖度のピークに集まり、密度が特定の基準を超過した場合に、対応する領域で細分化格子を持つ子ブロックが生成されることが確認できた(図7参照)。また、エージェントの動きは格子幅が変化するブロック境界においても妙な影響を受けず、AMR フレームワークへの粒子処理が適切に行われていることが確認された。

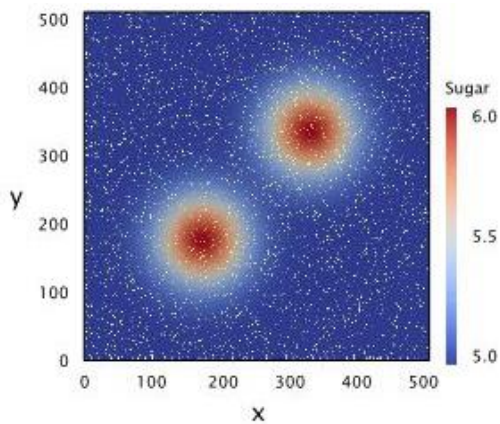


図6 計算領域内糖度分布

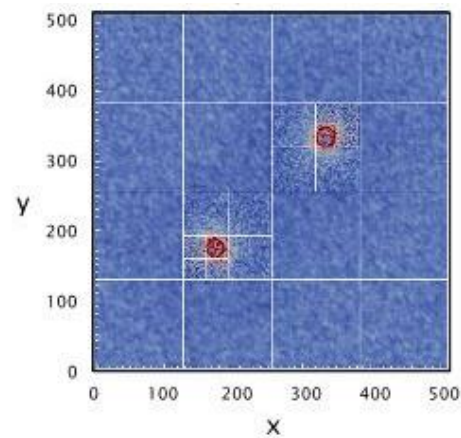


図7 エージェント密度とAMR ブロック分布



### 3.2 pK-Open-SOL の研究開発 (東大情報基盤センター・中島グループ)

#### (1) 目的

本研究の目的は悪条件な係数行列を持つ連立一次方程式を対象とした堅牢性の高いエクサスケールシステム向けのソルバの開発である。

本研究はドイツ SPEEXA プロジェクトの一つである ESSEX-II グループとの共同研究であり、同研究グループでは量子力学の大規模な一般固有値問題を解く事を目的としている。一般固有値問題の解法としては Arnoldi 法や Lanczos 法、Jacobi-Davidson 法などが広く知られているが、近年では Sakurai-Sugiura(SS)法や FEAST などが注目されている。SS 法や FEAST では任意の領域の周回積分を行うことで領域内の固有値を求めることができる手法であり、既存の手法よりも利便性が高いためである。ただし、周回積分によってシフト線形方程式群が導出され、悪条件な係数行列が含まれる可能性がある。加えて量子力学の一般固有値問題の係数行列は一般的に悪条件であるため、堅牢な大規模並列に対応した線形ソルバが必要とされる。実際に対象とする連立一次方程式の係数行列は以下の 4 つの性質のいくつかを同時に持つことを確認している。

- 1 行辺り数個～十数個の非零要素を持つ疎行列
- 非対角と比較して小さな対角成分の値
- 対角成分の値は正と負の両方
- 大きな条件数

#### (2) 実施方法

大規模な問題に対応した堅牢なソルバの開発を目的とするため、直接法と反復法の両方に関して検討を行った。

##### (2-1) 直接法

###### ① LU 分解

LU 分解に基づく解法は一般的な手法であり、堅牢性はここで検討した手法の中で最も高いと考えられる。ただし、対象の問題が疎行列であり、LU 分解後の行列は密行列となるため、メモリ使用量および演算量の観点から他の検討手法と比べて劣る。

###### ② Hmatrix を用いた LU 分解の近似

階層型行列法(Hmatrix)は密行列の近似手法として近年注目されている。本手法は対象の行列の部分行列に低ランク近似を適用することで、より少ないメモリ量で密行列を格納することができるためである。本検討では LU 分解後の密行列を Hmatrix を用いて近似することを考える。これにより、Hmatrix を用いない場合と比較してより大規模な問題を扱うことができる。ただし、効果的な低ランク近似を行うためには対象とする問題の座標情報などからクラスタリングを行う必要がある。

##### (2-2) 反復法

反復法は係数行列の疎行列性をそのままに近似解を求める手法である。一般的に直接法と比較して少ないメモリ量、演算量で解を求めることができる。反復法にはさまざまな手法が存在するが、ここでは前処理を併用することで堅牢性、収束性を改善できるクリロフ部分空間法を検討対象とする。

###### ① 前処理付きクリロフ部分空間法

###### (a) ILU 前処理

不完全 LU 分解は LU 分解を近似的に行う手法である。一般的な ILU 分解では、元の係数行列と同じ非零パターンになるように分解を行う。また、分解前の行列で 0 であった要素が分解の過程で非零になることを Fill-in と呼ぶ。Fill-in の発生を何らかの方策を基に許せば ILU 分解の結果がより完全な LU 分解に近くなり、より前処理の効果が高くなると期待できる。実際に、Fill-in の発生を任意のレベルまで許す ILU(l)や、非対角要素の値が任意の閾値以下になるまで Fill-in の発生を

許す ILU(t)などが提案されている〔原著 4〕。

#### (b) 代数マルチグリッド前処理

代数マルチグリッド(AMG)法は ILU 前処理と比較して少ない演算量で高い収束性改善が報告されている手法である。マルチグリッド法では任意の格子に対してより粗な格子を複数用意し、それぞれの格子空間で誤差を取り除く手法である。AMG 法で高い収束性改善効果を得るためには制約・補間演算子を対象の問題のカーネルを考慮して適切に決める必要がある。制約・補間演算子は密な格子と粗な格子の間の写像を行うために必要であり、本演算子によって粗な格子空間の効果が決まるためである〔原著 7,9,12,15,16〕。

#### (c) Hmatrix を利用した前処理

直接法の項で述べた Hmatrix を反復法の前処理として利用する手法である。直接法の場合との差はより精度を落とし演算量およびメモリ使用量を削減する点にある。クリロフ部分空間法の前処理では近似的に解を求めればよいため、直接法の場合よりもより低い精度で十分となる。本研究では LU 分解の過程でクラスタリングされた部分行列の形状を考慮した Fill-in を許可し、Hmatrix でこれを近似することを検討対象とする〔原著 13〕。

### (2-3) 検討

対象の問題の係数行列が疎行列であることから、反復法を用いた解法で解けることが、大規模化、性能の両観点から望ましい。Hmatrix や AMG を前処理として利用することを検討した結果、Hmatrix ではクラスタリングに利用する情報の推定が難しく、AMG 前処理では制約・補間演算子の算出方法の推定が困難である。本研究では ILU 前処理をより堅牢にし、エクサスケールを想定した並列化を施す手法を初めに採択した。本手法を適用した結果の詳細は後述する。結果として、特定の問題を除いて一般的な ILU 前処理付きクリロフ部分空間法で解けなかった問題を解けることを確認し、大規模並列環境で高い性能が得られることを確認した。また、提案した IC 前処理で解けなかった問題に関しては、SuperLU や MUMPS などの既存の直接法ライブラリに加え、独自の直接法などの適用を検討している。

### (3) 研究成果

エクサスケールを想定した堅牢な ILU 前処理開発のため、堅牢性向上を目的とした正則化および大規模並列化を目的とした多色順序付けの階層的な並列化手法の提案を行った。ここではそれぞれの成果について述べる。

#### (3-1) 正則化による堅牢な ILU 前処理の提案

ILU 分解を本研究で対象としている悪条件な問題、特に非対角成分に対して対角成分の値が小さい係数行列に適用した場合、計算誤差の増大や分解破綻を起こす可能性がある。そこで、本研究では 2 つの正則化を係数行列に適用し、対角成分の値を大きくした係数行列を導出、これに ILU 分解を施す手法を提案した。具体的に採用した正則化はブロック化と対角シフトの 2 つである〔口頭 16〕。

##### ① ブロック化

ブロック化を行った係数行列に ILU 分解を施すことで、対角ブロックの値がブロック化適用前と比較して大きくなり、堅牢性向上が期待できる。

前述のように対象とする問題は対角成分が非対角成分と比較して小さく、計算誤差が大きくなる可能性がある。ブロック化を元の係数行列に適用した場合、図 8 に示すように対角ブロックには非対角成分が含まれる。これにより非対角ブロックの値は元の係数行列と比較して大きくなるため、分解による計算誤差低減される可能性がある。また、ブロック ILU 分解ではブロック内の Fill-in 発生を許可するため、収束性改善の効果も期待できる。

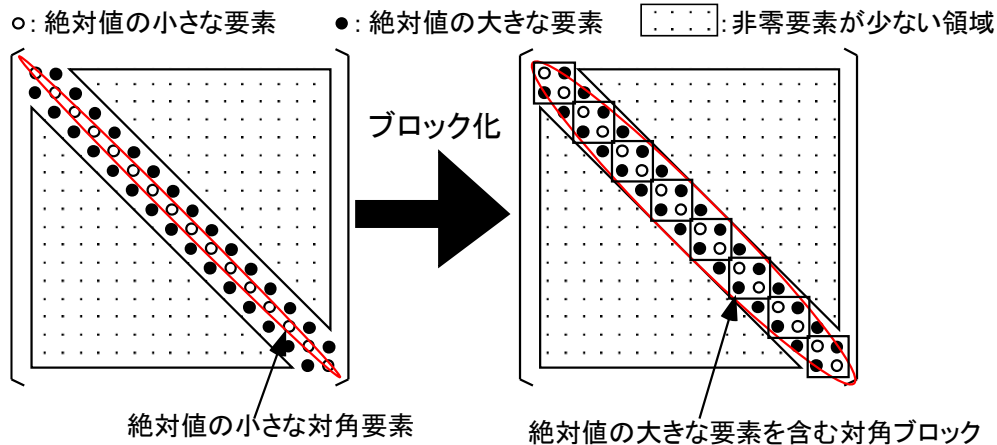


図 8 ブロック化の適用による対角成分の扱い

## ② 対角シフト

対角シフトは分解前の行列の対角成分を直接的に操作し、堅牢性を向上させる手法である。

本手法では任意の定数(シフト量)を対角成分に直接加算する。これにより対角成分を大きくなるため、計算誤差などを防ぐことが可能である。このような手法は電磁場解析などの分野で広く用いられており、複数の成果が報告されている。ただし、シフト量を大きくしすぎた場合、分解後の結果が元の係数行列と大きく異なってしまうため、逆に収束性を悪化させる可能性がある。したがって、電磁場解析の分野で実際に用いられるシフト量は対角成分の1~2倍程度が用いられる。対して、本研究で対象としている問題は電磁場問題よりも条件が悪い事が多く、同程度のシフト量では効果が得られなかったため、シフト量は10~100程度の加算を採用している。

## ③ 評価結果

ILU 前処理に提案した正則化を施すことで特定の問題を除いて高い効果を得られたことを示す。量子力学から導出される連立一次方程式を対象として ILU 前処理付きクリロフ部分空間法と正則化を施した手法との比較結果を示す。ESSEX-II グループより提供頂いた問題は Kohn-Sham、Graphen1、Graphen2、SpinSZ の4種類、合計で23ケースである。各問題の特徴を以下に示す。

### (a) Kohn-Sham

Kohn-Sham 方程式から原子同士、電子同士の干渉を考慮した電子軌道の解析を目的とした問題である。6 ケースのモデルが存在し、自由度は 57,575~76,163、非ゼロ要素数は1行辺り平均で20~24個である。

### (b) Graehene1、2

炭素分子のから構成される分子(カーボンナノチューブやフラーレンなど)の電気的特性の解析を目的とした問題である。Graphen1の自由度は1,000~1,000,000、1行辺りの非零要素数は13である。また、Graphen2の自由度は128~131,072、1行辺りの非零要素数は4である。

### (c) SpinSZ

電子スピンを分子構造内の相互作用も考慮して解析するための問題である。8 ケースのモデルが存在し、自由度は252~2,704,156、非ゼロ要素数は1行あたり6~12個である。

これらの問題から Kohn-Sham および Graphen1、2 の問題に対して検証を行った。対象の問題の係数行列は全て対称行列であるため、クリロフ部分空間法には CG 法を採用した。この時、ILU 前処理は IC 前処理と同義であるため、比較対象の手法は ICCG 法とする。

各問題を解く場合の右辺ベクトルは1~10の間の乱数とした。また、反復計算では初期残差に対する反復計算中の残差の相対値が  $10^{-7}$  以下となった場合に解が収束したと判断し、反復回数

が問題の自由度に達した場合には解けなかったと判断した。図9はブロックサイズを4、シフト量を100とし、ICCG法ブロック化のみ適用したICCG法、対角シフトのみ適用した手法、両方の正則化を適用した手法を比較した結果を示している。ICCG法では全15ケースのうち、9ケースで収束せず、1ケースで分解破綻となる結果となった。対して提案手法の適用により全15ケースの求解を達成した〔口頭16〕。

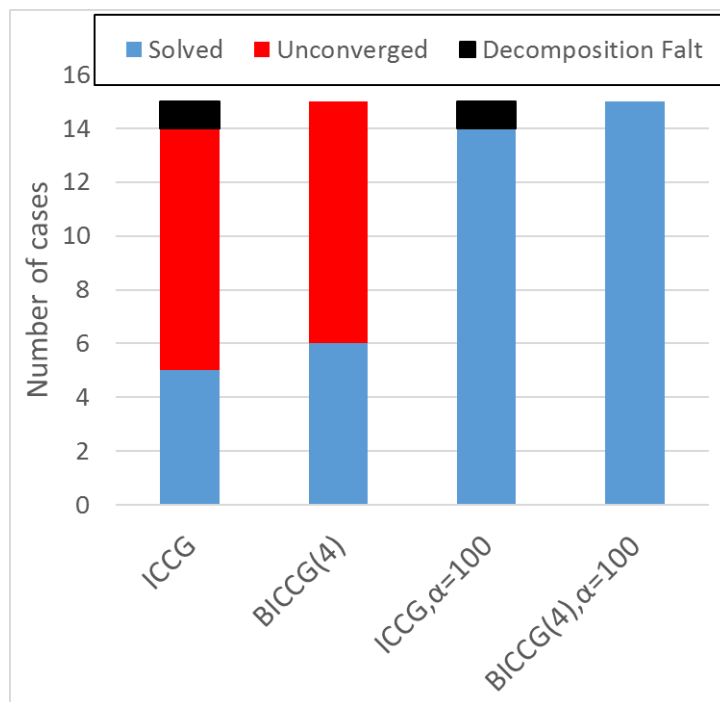


図9 正則化 ILU 前処理の効果

SpinSZの問題に対しては提案した正則化を施した場合でも解を得ることはできなかった。SpinSZとそれ以外の問題の大きな差は対角成分が正および負の両方を持っている点にあり、これが反復法で解きにくい要因になっていると考えられる。そこで、現在はさらに堅牢性の高いILU(t)および直接法の適用可能性について検討を行っている。これまでの成果ではMKLおよびSuperLUを用いてSpinSZのLU分解が可能であることを確認している。

### (3-2)階層的な並列多色順序づけ

ここではILU前処理の並列化手法として広く用いられている多色順序付けアルゴリズムの階層的な並列化について述べる。

ILU前処理のプロセスは主に分解と前進・後退代入に分別されるが、いずれも強い逐次性を持っている。これを並列化する手法として多色順序付けが広く用いられている。多色順序付けは対象の要素を要素間の依存関係に基づいて複数の色に色分けする手法である。この時、同じ色で塗られた要素間には依存関係がないように色分けされるため、ILU前処理の分解、前進・後退代入で同じ色の要素を並列に計算することが可能となる。ただし、ILU前処理の収束性は多色順序付けの結果に強く依存する。結果、Cuthill-McKeeやGreedy、代数多色順序付けなどさまざまな多色順序付けアルゴリズムが提案されており、多くのアプリケーションでの検証結果が報告されている。

一方で、既存の多色順序付けアルゴリズムは並列化されておらず、エクサスケールシステムで大規模な問題を解くためには不十分である。ILU前処理の効果が多色順序付けの結果に強く影響を受けることを想定した場合、既存の多色順序付けアルゴリズムを並列化可能な汎用的な手法が必要である。並列化された多色順序付けアルゴリズムはいくつか報告されているが、それらはグラフカラーリングのための物であり、ILU前処理での評価を行った報告は少ない。また、これらの並列化されたアルゴリズムは専用の手法であり、一般的にそれ以外の多色順序付けアルゴリズムを並列化

する物ではない。

本研究では階層的な手法に基づく既存の多色順序付けアルゴリズムを並列化可能な並列化の提案を行った。ここでは、逐次的な既存の多色順序付けアルゴリズムと提案手法で並列化されたアルゴリズムで ILU 前処理の効果に大きな差が出なかったことを示すとともに、1,000 ノード以上での動作実績について報告する。

#### ① 多色順序付けアルゴリズムの階層的な並列化

ここでは本研究で提案した多色順序付けアルゴリズムの並列化について述べる。

既存の多色順序付けアルゴリズムはある要素の色分け後、辞書式や Chutill-McKee 順序付けに基づいて次の色分け要素を決める。ここで、色分け後の要素と次の色分け要素が依存関係を持っていなければ、これらは並列に扱うことが可能である。提案手法ではそれぞれ依存関係がない要素グループを提供することで、それらを並列に色分け可能にする。

図 10 に提案手法の手順を示す。具体的な手順は以下の通りである。

- a 各プロセスに担当する要素を割り当てる
- b 各プロセスは担当の要素を複数のグループに分割する。
- c 分割されたグループの依存関係から新たなグラフを作成し、これをマスタープロセスに送付する。
- d マスタープロセスが集約されたグラフに基づいて任意の手法で色分けを行う。
- e 色分け結果を他のプロセスに送付する。
- f ~ h 各プロセスはグループの色分け結果に基づき並列に任意の多色順序付けを行う。

グループ毎の色分けは任意の多色順序付けアルゴリズムが適用可能であるため、既存の多色順序付けアルゴリズムを並列化可能な手法である。

#### ② 評価結果

提案手法の評価結果を以下に示す。図 11 は逐次の多色順序付け法に対して階層的な並列化を施した多色順序付け法の収束性と計算時間の変化を示している。評価した多色順序付け法は Greedy と  $\alpha$  色の代数多色順序付け(AMC( $\alpha$ ))、およびそれらと Cuthill-McKee(CM)と併用した手法である。各グラフにおいて反復回数が 1 であれば逐次的な多色順序付けと階層的な並列化を施した多色順序付けの反復回数が同じであり、1 より大きい場合は収束性が良いことを示す。計算時間のグラフに関しても同様である。なお、計算時間は反復部のみの測定である。

対象とした問題は Florida matrix collection の Parabolic\_FEM、Thermal2、Flan1565 の 3 つおよび 3 次元ポアソン方程式を構造格子を基に有限差分法で離散化した問題(自由度は  $128 \times 128 \times 256$ )として評価を行った。反復法はブロックサイズ 4、シフト量 0 の正則化 ICCG 法を適用し、相対残差が  $10^{-7}$  以下となった場合に収束と判断した。評価に用いた環境は東京大学 情報基盤センターの Reedbush-U の 32 ノードである。ブロック ICCG 法はハイブリッド並列化されており、ノード辺り 2 プロセス、プロセス辺りのスレッド数は 18 とした。

いずれの問題でも階層的な並列化を適用した多色順序付け法の収束性および計算時間は逐次の場合と比較して同程度であった。最も収束性の差が大きい場合で 1.32 倍(Parabolic\_FEM の Greedy)、計算時間では 1.26 倍(FLAN1565 の Greedy)であった。

次に、ノード数を 1~128(ノード辺り 2 プロセス、プロセス辺り 18 スレッド固定)で変化させた場合の結果を図 12 および表 1 に示す。図 12 はノード数を変化させた場合の台数効果を、表 1 はノード数に対する収束性の変化を示している。対象の問題は独自のポアソン問題( $512 \times 512 \times 1024$ )としている。多色順序付けのアルゴリズムには 10 色の代数多色順序付けを用いている。階層的な並列化を適用により、ノード数の変化に対して色分け結果は変化するが、収束性および計算時間への影響は小さい。これらの結果から、提案手法は既存の多色順序付けの特性を変化させずに並列化可能な手法であることが確認された[原著 20、口頭 24]。

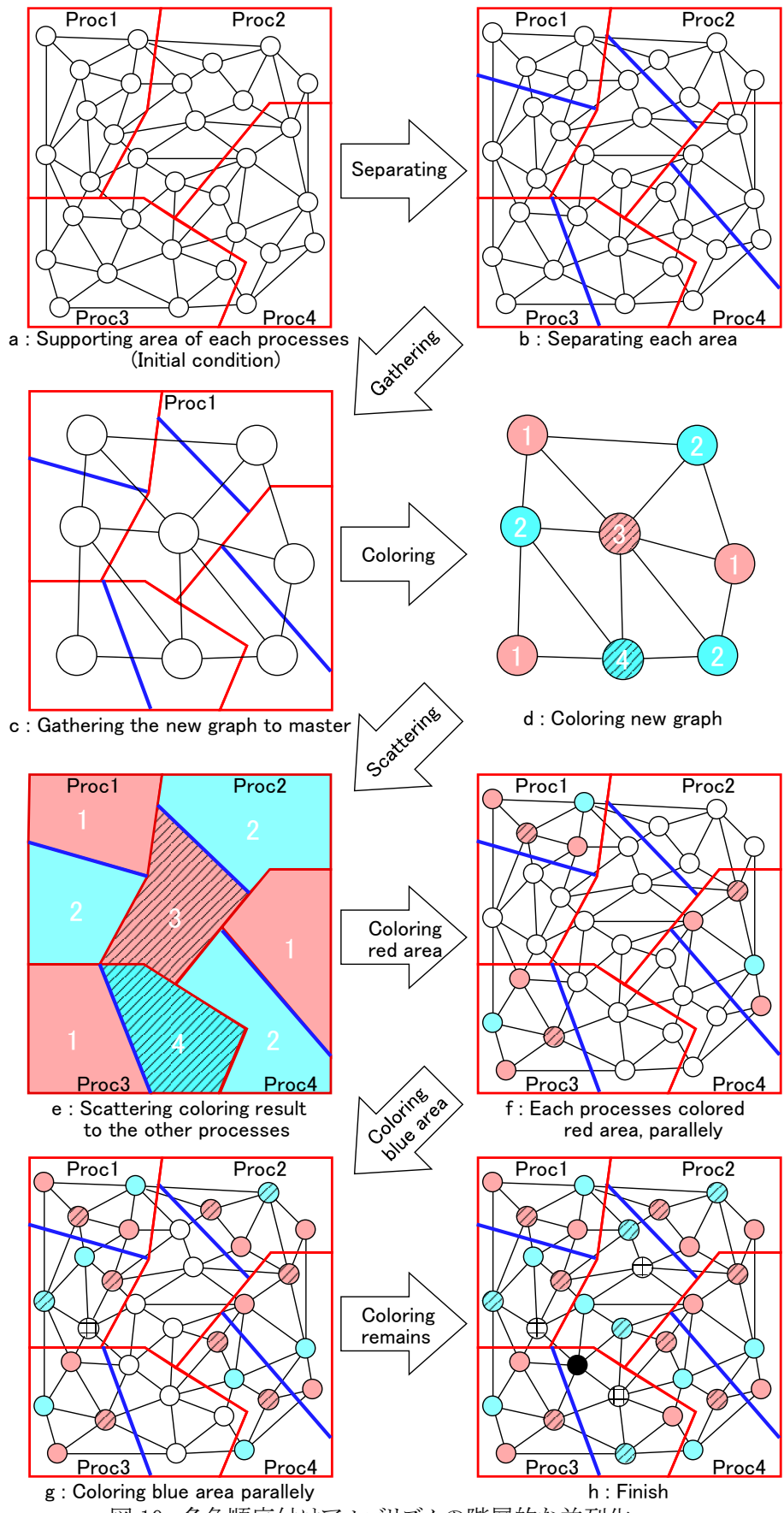


図 10 多色順序付けアルゴリズムの階層的な並列化

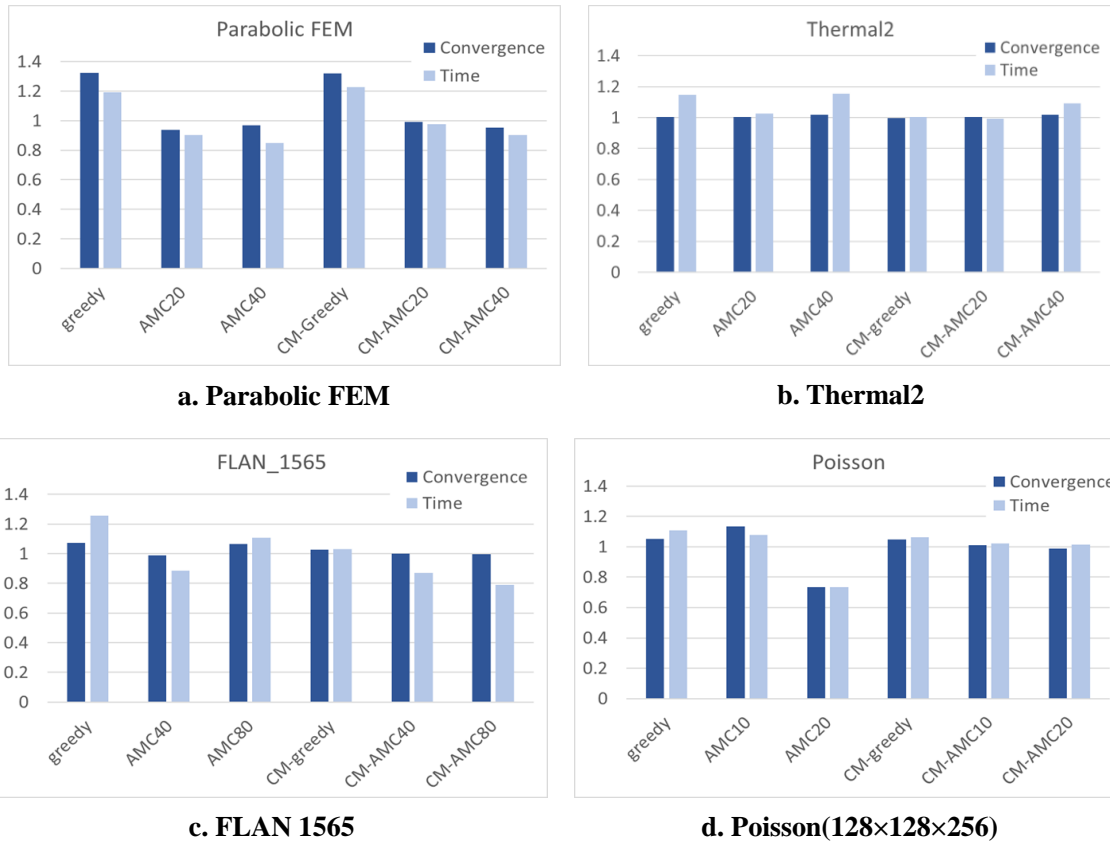


図 11 逐次多色順序付けに対する階層的な並列化の影響

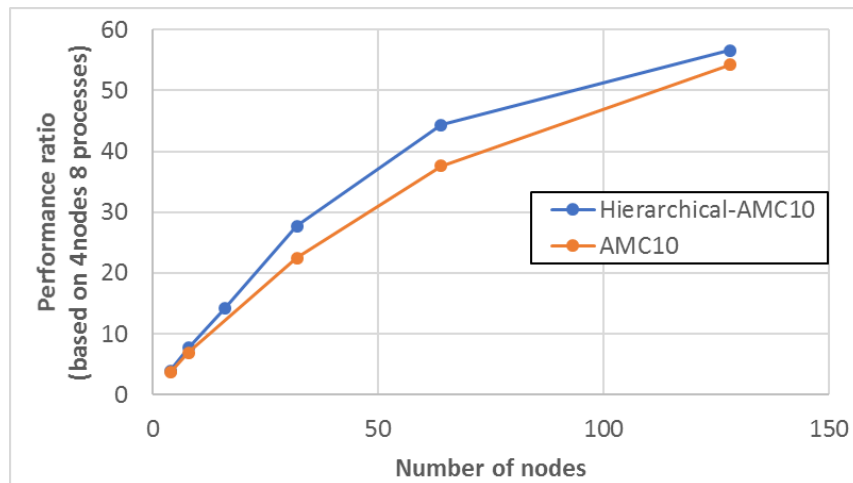


図 12 Poisson(512×512×1024)での並列化の効果

表 1 Poisson(512×512×1024)でのノード数に対する反復回数の変化

		ノード数					
		4	8	16	32	64	128
反復回数	並列多色順序付け	1,964	1,936	1,937	1,969	1,824	1,957
	逐次多色順序付け	2,194					

最後に、より大規模な並列環境での解析結果を示す。対象は前節で使用した Graphene モデルのさらに大規模(約 3,300 万自由度)な問題である。評価には東京大学学術情報基盤センターの Oakleaf-FX 4~1,024 ノードを利用した。本評価ではノード辺り 16 スレッドのハイブリッド並列としている。また、色分けには提案手法で並列化した 10 色の代数多色順序付け法を使用している。ノード数に対する性能向上を**エラー! 参照元が見つかりません。**に、反復回数の変化を**エラー! 参照元が見つかりません。**に示す。並列度を 1,024 ノードまで増加させた場合でも反復回数の変化に大きな差はなく、十分な性能を確認した。また、**エラー! 参照元が見つかりません。**は同問題での並列多色順序付け法の性能評価結果を示している。反復法部と同様、1,024 ノードでも良好な性能を確認した。なお、多色順序付けに要する時間は反復計算部のおおよそ 0.3%程度である。

これらの結果から、提案した多色順序付けの階層的な並列化手法は効果的であり、エクサスケールシステムでも十分な効果が得られると判断する。

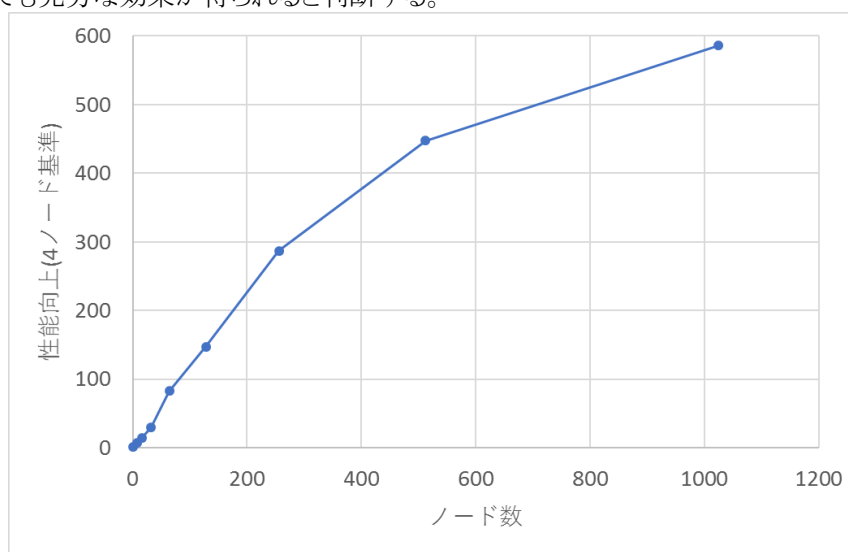


図 13 Graphene8194×4096 の FX10 上での評価結果

表 2 Graphene8194×4096 の反復回数の変化

	ノード数								
	4	8	16	32	64	128	256	512	1024
反復回数	27,812	27,823	27,812	27,823	27,801	27,823	27,823	27,801	27,823

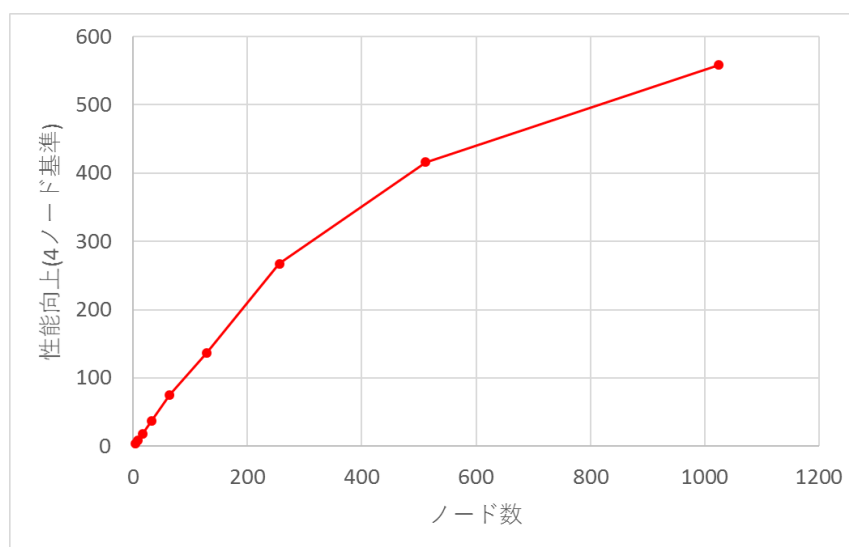


図 14 Graphene8194×4096 での並列多色順序付けの性能評価



#### (4)ソフトウェアの公開状況

本研究で開発したプログラムは ESSEX-II で公開されている PHIST ライブラリに実装し、公開する予定である。

PHIST には Jacobi-Davidson や FEAST などの固有値ソルバが実装されており、ESSEX-II で同様に公開されている GHOST や MKL、CuBLAS など複数の数学ライブラリを対象の環境に応じて切り替えることを想定したソフトウェアである(エラー! 参照元が見つかりません。)。PHIST に含まれる FEAST などから導出される連立一次方程式のソルバとしては GMRES や BiCGStab などの一般的なソルバに加えて、Carp-CG と呼ばれる量子力学の問題を解く事を目的としたソルバが含まれている。pK-Open-SOL の成果はこの Carp-CG と同じレイヤーへの実装を進めている。具体的な手順は CRS からブロック CRS への変換関数や行列ベクトル積などクリロフ部分空間法の実装に必要な計算を行うための関数の実装(Built-in Block version)および、ILU 分解・前進後退代入計算、並列多色順序付けを行う関数の実装である。PHIST への実装という形で公開することにより、ヘテロジニアスな環境を含む様々なシステム上での高い性能を実現する。

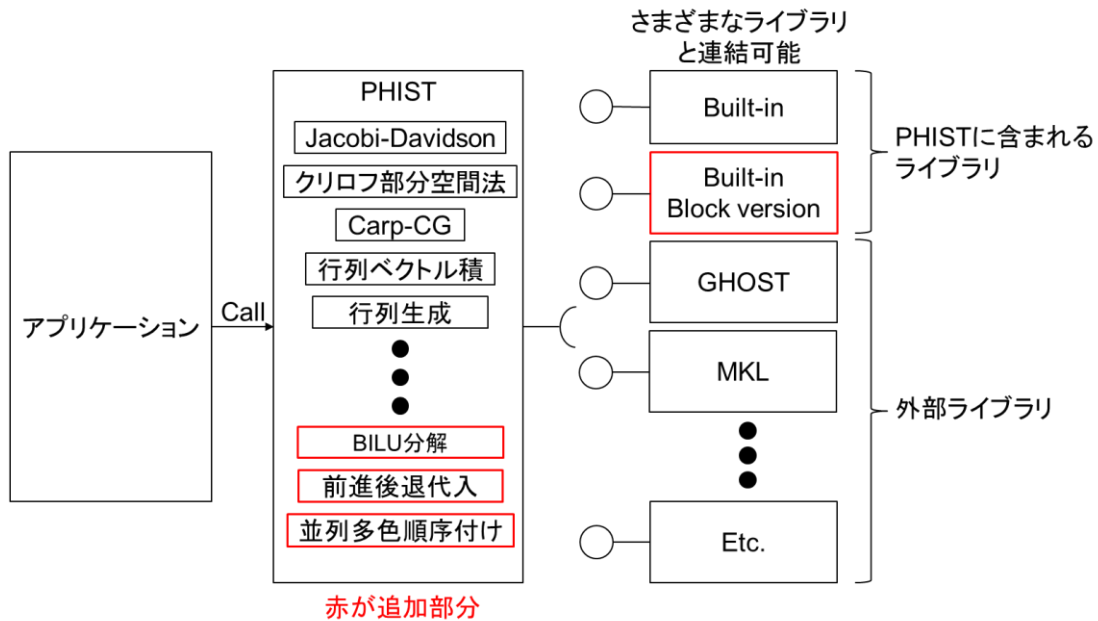


図 15 PHIST への実装 概念図

#### (5)エクサスケールへの貢献

本研究で提案した堅牢性を向上させた ILU 前処理および階層的な並列化多色順序付け法は、求解が困難な大規模問題をエクサスケールシステムで解けるようにするという観点で大きくエクサスケールに貢献する物である。

一般的に ILU 前処理を分散メモリ環境で実装した場合、Localized ILU が採用されることが多い。ILU 前処理の分散メモリ環境に対応した実装は非常に困難であり、また多色順序付けの逐次性という問題があるためである。ただし、Localized ILU では並列度の増加とともに収束性が低下する問題がある。研究成果の項で述べたように開発したソルバは並列度に対して収束性が変化しないため、本研究で対象とするような悪条件かつ大規模な問題のエクサスケールシステム上での求解に適している。

### 3.3 pK-Open-AT の開発(東京大学 中島グループ)

#### (1) pK-Open-AT の概要

##### (1-1) pK-Open-AT の機能

計算機ハードウェアの性能に自動的に適合し、ソフトウェアを自動的に高速化するパラダイムである自動チューニング機能(AT 機能)について研究開発を行った。ppOpen-HPC プロジェクトで開発された ppOpen-AT をもとに、pK-Open-AT を開発し、特にメモリが多階層化する際に必須となる「階層型 AT 機能」の研究開発、および、新規にリリースされた計算機環境を含む最先端のCPUにおける AT 効果の検証を行った。対象となるアプリケーションは、有限差分法(FDM)を基にした陽解法のプログラムである ppOpen-APPL/FDM である。

##### (1-2) pK-Open-AT の特徴

AT による性能チューニングに関し、先行研究である pp-Open-HPC では、以下のコード変換による AT 機能を実装している:

- ① レジスタ最適化のためのループ分割(Loop Split)、および式の並び替え
- ② スレッド最適化のため、ループ長を長くするコード変換であるループ融合(Loop Collapse)
- ③ 低 Byte per Flops (B/F) となる専用カーネル 2 種と、そのコード切り替え

以上の機能、特に③の低 B/F となる専用カーネルとの切り替え機能を AT 機能として有する AT 方式について FDM コードに実装されているのは、2017 年現在 ppOpen-APPL/FDM のみである。この観点で、独創性を有する。開発した機能は階層型メモリへの適用を考慮し、上記の最適化を「下位のメモリ」向けコード最適化と位置づける。ループ分割などのコード最適化により、キャッシュより下位のメモリ(1次、2次キャッシュ、およびレジスタ)を対象とした最適化と位置づける。

一方、「上位のメモリ」を、最下層キャッシュ(LLC)および3次元積層メモリなどの高バンド幅メモリと定義する。このとき、この上位のメモリで主に行われるチューニングは、アルゴリズムレベルの最適化(コード最適化)であるとする。この仮定による、「上位のメモリ」向けコード最適化をコード選択とする。

以上で示した上位コードの最適化指定(コード選択)は、AT 専用言語 ppOpen-AT を拡張した pK-Open-AT での“select”構文を利用し指定を行うことができる。一方、下位コードの最適化指定(レジスタ最適化など)は、上位コードから呼ばれる手続き内の処理に対する AT 指定になる。

##### (1-3) pK-Open-AT を実行可能な計算機環境

開発コードは、Fortran コンパイラと Message Passing Interface (MPI)が搭載されている計算機で動作する。利用開始された直後の計算機においても、コードの変更無しで AT 機能が動作する。そのため、あらかじめ実装された AT 方式における機能の範囲内で、自動的な高速化が達成可能となる。以上のことから、GPU 以外の計算機環境で実行可能である。

##### (1-4) pK-Open-AT の潜在的ユーザ

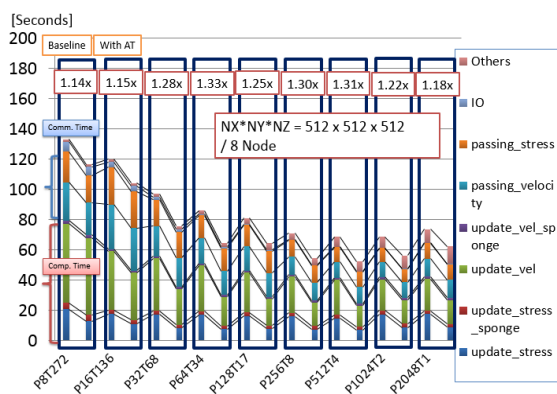
以下のユーザの利用を想定している:

- ① ppOpen-APPL/FDM をブラックボックスツールとして利用するユーザ
- ② 新しい計算機において、どのような最適化方法が有効となるか知りたいユーザ
- ③ AT 方式の研究者

①のブラックボックスツールとして利用するユーザは、実行速度の観点での利益を享受する。それ以外の潜在的ユーザとして、2の ppOpen-APPL/FDM で採用されている計算方式(例えば、4次の差分スキーム)を採用しているユーザについて、AT結果の最速実装方式(例えば、ループ融合

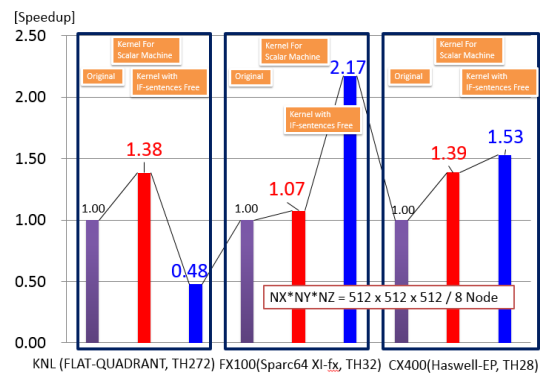
の仕方など)を参照することで、自分のコードの最適化にAT結果の知見を活用できる。また、3 のループ変換を研究するAT方式の研究者の利用も想定できるため、幅広い潜在ユーザが存在する。コード選択の対象処理を、開発した以下の3種とした:

- ① **従来コード**: B/F 値が約 1.7 と高いコードである。経験的に、ベクトル計算機で最も高い性能を発揮するように実装されている。
- ② **ループ融合コード (IF 文あり)**: 従来コード中の処理である、(i)4 次の中央差分スキーム、(ii)領域境界処理、(iii)leap-frog スキームによる時間発展、をループ内に記載したコードである。配列への入出力回数を減らすことで、B/F 値を約 0.4 まで削減した。(ii)領域境界処理のため、IF 文が最内ループに存在する。
- ③ **ループ融合コード (IF 文なし)**: 2 のコードにおいて、(i)IF 文を取り除いたループ、(ii)境界処理専用ループ (IF 文あり)、の 2 つのループに分離したコードである。(i)のループは IF 文が無いいため、コンパイラによるコード最適化の適用が期待できる。



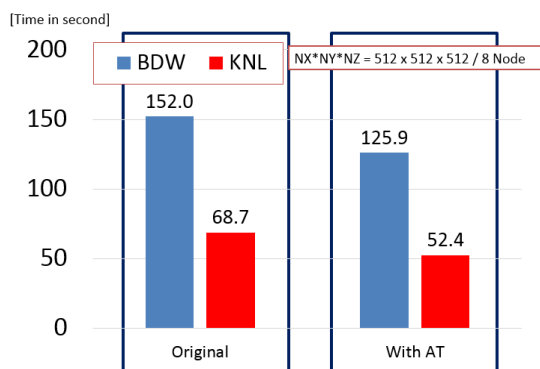
(a) KNL における AT 効果

(FLAT-QUADRANT、8 ノード利用時。なお、ハイパースレッド数は 4 である。)

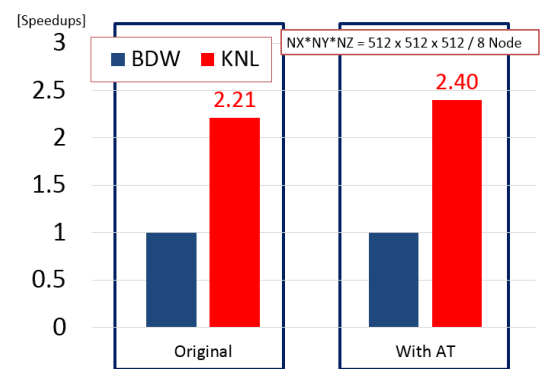


(b) KNL、FX100、および Haswell-EP

における AT 効果(カーネル update\_stress)。左から、オリジナル実行、2 ループ融合コード (IF 文あり)、ループ融合コード (IF 文なし) の実行時間である。



(c) BDW と KNL の最速実行時間



(d) BDW と KNL における最速実行時間の BDW の時間を 1 としたときの KNL の速度向上率

図 16 AT 効果および実行時間の比較。

当該カーネルの 2000 時間ステップまでの経過時間の蓄積時間に対する値である。

以上のコード選択機能付きの AT 方式を実装して性能評価を行った。ppOpen-APPL/FDM を、東京大学と筑波大学で構成される JCAHPC が導入した Oakforest-PACS のノードである Intel Xeon Phi (Knights Landing, KNL)、東京大学情報基盤センターの Reedbush-U のノードである Broadwell EP(BDW)、および名古屋大学情報基盤センターの Fujitsu PRIMEHPC FX100 のノードである Sparc64 Xifx (FX100)、および Fujitsu CX400 のノードである Haswell-EP で AT の効果を検証した。ここで、KNL および FX100 のノードは、3次元積層技術による最新鋭メモリが搭載されているため、双方の実効バンド幅は 300GB/秒程度に達する。図 16 に結果を示す〔原著 2,8〕。

図 16 (a)では、KNL のメモリモードおよびコア割り当てモードの 1 つである FLAT-QUADRANT モードの実行結果である。X 軸はハイブリッド MPI/OpenMP の実行形態であり、例えば P8T272 は、8MPI プロセスで、それぞれの MPI プロセスから 272 スレッドが生成される実行形態である。図 16 (a) から、AT による全体性能の速度向上率は最大で 1.33 倍に及んでいる。

図 16 (b)は、カーネル update\_stress の KNL、FX100、および Haswell-EP における AT 効果である。図 16(b)の結果から、KNL では「ループ融合コード(IF 文あり)」が高速であるが、それ以外の CPU では、「ループ融合コード(IF 文なし)」が高速であることがわかる。このように計算機に依存して最適な実装方式が変わることが確認されているため、本提案による AT 方式は、最先端の CPU による異機種環境でも有効に機能することが実証された。

図 16(c)(d)は、AT あり/なしについて、BDW と KNL の最速実行時間と最速実行時間について BDW の時間を 1 としたときの KNL の速度向上率を示している。図 16(d)から、AT ありのほうが BDW に対する KNL の速度向上率が高く、約 2.4 倍を達成している。この観点においても、提案する AT 方式の有用性が示されている。また、KNL の BDW に対するバンド幅の向上率は約 3 倍であること、および ppOpen-APPL/FDM はメモリバンド幅律速なコードであることを考慮すると、まだチューニングの余地があるものの、KNL ではハードウェアによる性能向上に対して妥当な性能を達成していることがわかる。

以上のように、本研究で開発した FDM のための AT 方式は、リリースされた直後の計算機環境においても有効に機能し、AT による速度向上が達成できることが実証できた。

### (3) エクサスケールへの貢献

エクサスケール環境における計算機は多様となることが予想されているが、本年度実証した計算環境 (Intel Xeon Phi (KNL)) は 2017 年現在の最新鋭のマルチコア環境である。そのため、最新鋭のマルチコア環境における高性能化を、自動的に、かつ低コストで実施できることを実証したことから、この観点においてエクサスケールへの貢献が期待できる。

一方で、マルチコア環境においても、実装したループ変換による AT 方式は有効である。また、ベクトル計算機においては、従来コードが対応しているため、コード選択により自動的に従来コードが選択される。そのため、多様なエクサスケール環境においても、性能可搬性を達成できる方式であることが期待できる。

本研究で開発した AT 方式は、ループ変換を主とする汎用的なコード最適化手法を基にしている。FDM を採用している他分野のコード最適化にも適用できる AT 方式であるといえる。2017 年現在、プラズマ・核融合コードにおいても、本研究で開発した AT 方式が適用可能であることが指摘されている。そのため、幅広い計算科学のコードに対して適用可能であることが期待される。

## 3.4 クリロフ部分空間法における通信最適化(東京大学 中島グループ)

### (1) 概要

大規模な並列計算機を使用する場合、ノード数の増加によって通信のオーバーヘッドは増加する傾向にある。並列計算において通信は必須のプロセスであるが、通信をできる限り効率的に実施し、削減することはエクサフロップス級システムにおいて重要である。通信の削減がアルゴリズムに不安定をもたらす可能性もある。ここでは、クリロフ部分空間法に焦点を当て、①パイプライン法、

②動的ループスケジューリングに関する試みを紹介する。本節におけるコード群は公開していない。

## (2)パイプライン法

有限要素法, 差分法等から得られる大規模疎行列の解法としては共役勾配法 (Conjugate Gradient, CG) に代表されるクリロフ部分空間法が広く使用されている。共役勾配法では内積の計算が現れるが, 全MPIプロセスによる通信を必要とするため, 計算量に比べて通信のオーバーヘッドの割合が大きくなる。Ghysels等によるPipelined CG法〔口頭14〕では, CG法の結果が変わらないように計算順序を変更し, MPI-3でサポートされている非同期集団通信 (MPI\_lallreduce) によって, 集団通信のオーバーヘッドを隠蔽することに成功している。〔口頭14〕では三次元弾性問題に有限要素法を適用して得られる対称正定行列を係数行列とする連立一次方程式を, 局所化されたブロックヤコビ型Symmetric Gauss Seidel (SGS) 法を前処理として適用したCG法 (LSGS-CG) によって解いている。以下に示す4種類のアルゴリズムについて検討を実施した:

- Alg.1: オリジナルのLSGS-CG法
- Alg.2: Chronopoulos/GearアルゴリズムによるLSGS-CG法〔口頭14〕
- Alg.3: Pipelined CG法〔口頭14〕
- Alg.4: Groppアルゴリズムによる CG法〔口頭14〕

計算にはReedbush-Uシステム (RBU, 東京大学情報基盤センター) を使用した。計算の概要は表1に示す通りである。Reedbush-U 1ソケットの18コアのうち16コアを使用し, 最小2ノード (4ソケット, 64コア) から最大384ノード (768ソケット, 12,288コア) までのStrong Scaling性能を評価した。

表 3 Reedbush-U における計算の概要

問題規模	Small: 256 × 128 × 144 節点 (14,155,776 DOF) Medium: 256 × 128 × 288 節点 (28,311,552 DOF)
利用ノード数	最小 2 ノード, 4 ソケット 最大 384 ノード, 768 ソケット 1 ソケットあたり 16 コア使用, 最大 12,288 コア Small: コア当り最小問題サイズ = 1,152 DOF/core Medium: 同 = 2,304 DOF/core
並列プログラミングモデル	Flat MPI: 1 コアあたり 1 MPI プロセス Hybrid: 1 ソケットあたり 1 MPI プロセス, 16 スレッド
ノード内リオーダーリング法	RCM
ASDD 反復数	2 回
CG 法収束条件	$ r / b  =  b - Ax / b  < 1.0 \times 10^{-8}$

図 17 CG 法の反復回数とコア数の関係

まず、図 17 にコア数と反復回数の関係を示す。ブロックヤコビ型局所 SGS 前処理を適用しているため、コア数が増加すると反復回数が増加する傾向があるものの、ASDD (加法型 Schwartz 領域分割法) の効果によって Hybrid の場合は 64 コアから 12,288 コアまで 10% 程度の反復回数増加に留まっているが、Flat MPI の場合は 20% 以上となっている。図 18 は Hybrid の 384 ノード、12,288 コアまでの速度向上率である。Flat MPI 2 ノード (4 ソケット、64 コア) の Alg.1 の性能を 64.0 としている。Small, Medium のいずれの場合にも Alg.1 の速度向上率が低く、Alg.2 がややそれより良く、Alg.3, Alg.4 は更に良く、両者の速度向上率はほぼ同じである。Medium での Alg.3, Alg.4 の速度向上率は 12,888 コアで 9,000 を超えており、約 75% の並列化効率が得られている。

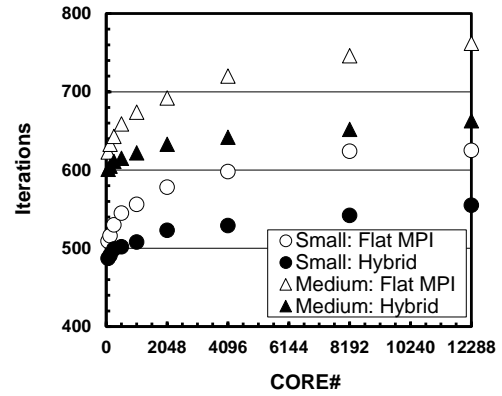


図 19 は Alg.3, Alg.4 において非同期集団通信関数 (MPI\_Iallreduce) を使った場合 (IAR) と通常の MPI\_Allreduce を使った場合 (AR) と Alg.1 を比較したものである。非同期集団通信関数を使用しない場合は、Alg.1 とほぼ同じ挙動を示していることがわかり、非同期集団通信機能の適用により、通信と計算のオーバーラップを実現し、大幅な性能向上が得られていることがわかる。

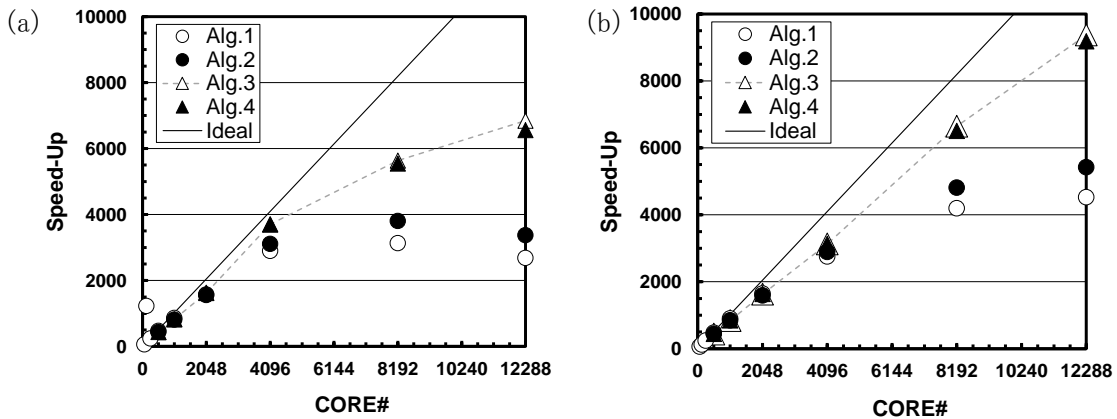


図 18 並列 CG 法の計算性能 (Hybrid), Flat MPI 2 ノード (4 ソケット、64 コア) (Alg.1) の性能を 64.0 としたときの速度向上率 (a) Small, (b) Medium

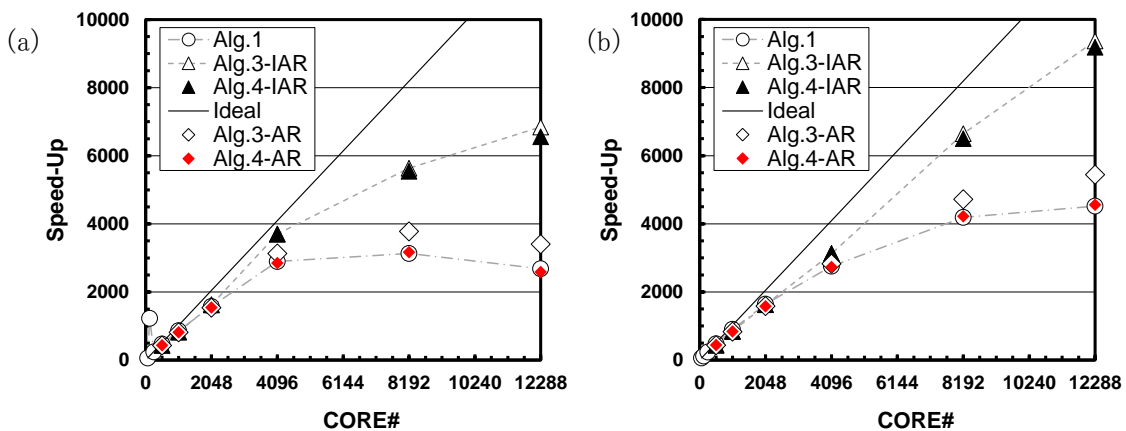


図 19 並列 CG 法の計算性能 (Hybrid), Flat MPI 2 ノード (4 ソケット、64 コア) (Alg.1) の性能を

64.0 としたときの速度向上率 (a) Small, (b) Medium, IAR:MPI\_Iallreduce 使用, AR:MPI\_Reduce 使用

### (3) 動的ループスケジューリングによる通信最適化

クリロフ部分空間法において通信を必要とするプロセスとしては、内積のほかに疎行列ベクトル積 (Sparse Matrix Vector Multiply, SpMV) がある。SpMVにおける通信最適化の手法としては、計算と通信のオーバーラップ (Computation Communication Overlapping) [原著14] が広く知られている。図20に示すように、並列有限要素法、有限体積法などにおけるHaloを含む並列データ構造において、内点を、Haloと接している点 (境界点) とそうでない点 (純内点) にわけて並び替えることによって、後者の計算とプロセス間をオーバーラップさせることができる。図20に示す手法は、本来ステンシル計算 (陽解法) のために考案されたものであるが、疎行列ベクトル積の場合には、通信を隠蔽するだけの十分な計算量を得ることが難しく、あまり効果がないとされてきた。一方、Halo通信にOpenMPの動的ループスケジューリング機能を適用する手法は、通信をマスタースレッドが受け持ち、純内点の計算とオーバーラップさせる。純内点の部分の計算には、OpenMPの動的ループスケジューリング機能を適用し、適切なチャンクサイズを設定し、通信終了次第マスタースレッドが計算に参加できるようになっている。

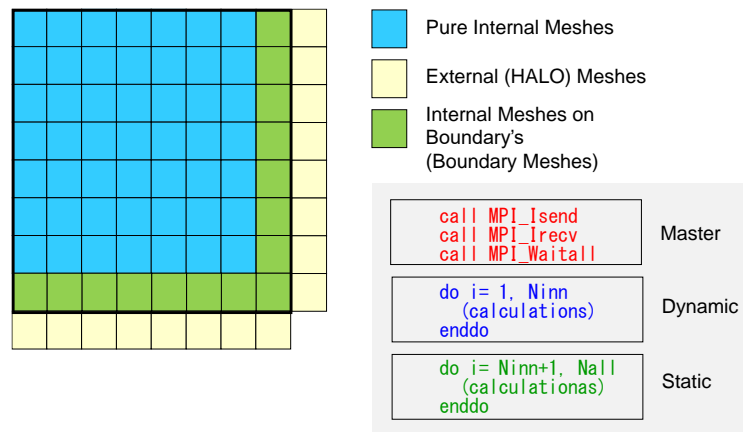


図 20 並列 SpMV における計算と通信のオーバーラップ

```

!$omp parallel private (neib,j,k,i,X1,X2,X3,WVAL1,WVAL2,WVAL3)
!$omp& private (istart,inum,ii,ierr)
!$omp master      Communication is done by the master thread (#0)
!C
!C- Send & Recv.
(...)
call MPI_WAITALL (2*NEIBPETOT, req1, stal, ierr)
!$omp end master

!C
!C-- Pure Internal Nodes      The master thread can join computing of internal
                             nodes after the completion of communication

!$omp do schedule (dynamic,200)  Chunk Size= 200
do j= 1, Ninn
(...)
enddo

!C
!C-- Boundary Nodes          Computing for boundary nodes are by all threads
                             default: !$omp do schedule (static)

!$omp do
do j= Ninn+1, N
(...)
enddo

!$omp end parallel

```

図 21 並列 SpMV への動的ループスケジューリング適用

表 4 使用した計算機システムの各ソケット・コアの性能 [原著 14]

System	Oakleaf-FX (FX10)	Reedbush (RBU)	Oakforest-PACS (OFP)
Name in this Paper	FX10	BDW	KNL
Architecture	Fujitsu SPARC64 IX fx	Intel Xeon E5-2695 v4 (Broadwell-EP)	Intel Xeon Phi 7250 (Knights Landing)
Frequency(GHz)	1.848	2.10	1.40
Core #/CPU (socket) (Max Effective Thread #)	16 (16)	18 (18)	68 (272)
Peak Performance (GFLOPS)	236.5	604.8	3,046.4
Memory Size(GB)	32	128	MCDRAM: 16 DDR4: 96
Memory Bandwidth(GB/sec, STREAM Triad)	64.7	65.5	MCDRAM: 490 DDR4: 84.5
Peak Performance/core (GFLOPS)	14.8	33.6	44.8
Memory Bandwidth/core (GB/sec., STREAM Triad [22])	4.04	3.64	MCDRAM: 7.21 DDR4: 1.24

本研究[原著 14]では、OpenMP/MPI ハイブリッド並列プログラミングモデルを使用した有限要素法による三次元固体力学アプリケーション向け前処理付きCG法(対角ブロック前処理)における疎行列ベクトル積に本手法を適用した(図 21)。通信をマスタースレッドが受け持ち、他のスレッドは「`omp do schedule (dynamic,CHSIZ)`」(CHSIZ:チャンクサイズ)として実施した計算結果を図 22 に示す。Staticは通常の「計算-通信オーバーラップ(動的ループスケジューリング適用無し)」であり、50, 100, ..., 500 などの数字はチャンクサイズである。東大情報基盤センターOakleaf-FX (FX10), Reedbush-U (BDW), Oakforest-PACS (OFP) の 3,840 コア (FX10:240 ノード, BDW:120 ノード, OFP:60 ノード) を使用している(表 4 参照)。問題サイズは 122,880,000 節点 (640×480×400), 368,640,000 DOF である。図 22 の結果は、オリジナル実装からの性能改善率であり、FX10:10-15%, KNL:20-25%程度 of 大幅な性能向上が得られている。一方、BDW では性能改善率は 5%程度である。一つの原因として、BDW (Reedbush-U) のノード間通信性能が、FX10 (Oakleaf-FX), KNL (Oakforest-PACS) と比較して良いことが考えられる。また、動的ループスケジューリングを適用しない通常の計算・通信のオーバーラップ (Static) はほとんど性能向上がない。最適なチャンクサイズは、問題サイズ、ノード数、機種によって異なると思われるため、自動チューニングの適用が効果的である。

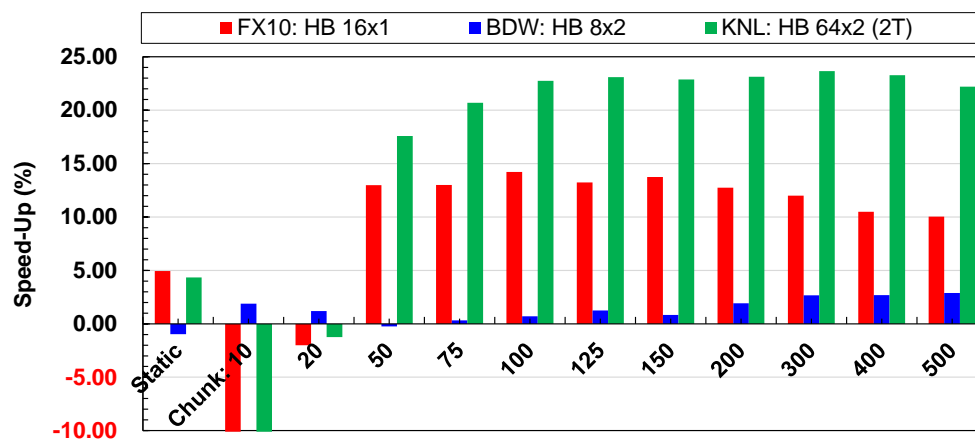




図 22 動的ループスケジューリングの効果(三次元固体力学並列有限要素法, 対角ブロック前処理付き CG 法), オリジナル手法からの CG 法性能改善率, Static:動的ループスケジューリング適用無し, 数字はチャンクサイズ, FX10 (Oakleaf-FX), BDW (Reedbush-U), KNL (Oakforest-PACS) 3,840 コア, 122,880,000 節点 (640×480×400), 368,640,000 自由度

### 3.5 Fault Resilient な並列計算のための動的負荷分散手法(東京大学 中島グループ)

ESSEX-II プロジェクトでは, エクサフロップス級システムにおいて Check-Point に基づく Fault Tolerance を実現するライブラリとして, CRAFT (A library for application-level Checkpoint/Restart & Automatic Fault Tolerance) を開発している。ppOpen-HPC/pK-Open-HPC の動的負荷分散機能を使用して, 超並列環境において, 有限要素法等のシミュレーションを実行中に計算ノードにハードエラーが生じた場合に, いわゆるスペアノードを使用することなく, 残りの計算資源を使用してシミュレーションを継続するフレームワークを開発中である。図 23 にその概念図を示す。8 プロセスで計算を実行中に, (a) PE#3, PE#5 が故障した場合, (b) まず PE#0, PE#1 で故障ノードが担当していたデータ(形状ファイル, チェックポイントファイル)を読み込み, (c) ppOpen-APPL/FVM の動的負荷分散機能によって, 各プロセスの負荷を均等にし, (d) 残りの計算資源で計算を継続する, というアイデアである。

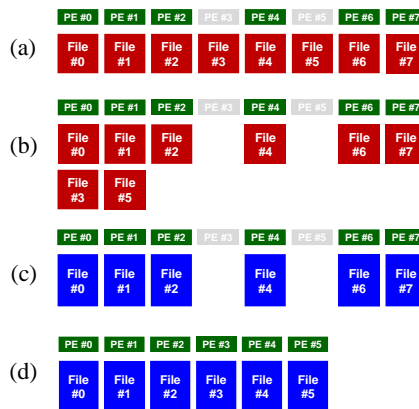


図 23 ppOpen-APPL/FVM による Fault-Resilience の実現

## § 4 成果発表等

(1) 原著論文発表 (国内(和文)誌 1 件、国際(欧文)誌 15 件)

〈国内〉

1. 松本正晴, 奥野喜裕, パルスレーザー駆動 MHD 発電の実証基礎研究, 日本航空宇宙学会論文集 62(2), 91-96, 2016

〈国際〉

2. Takahiro Katagiri, Masaharu Matsumoto, Satoshi Ohshima: Auto-tuning of Hybrid MPI/OpenMP Execution with Code Selection by ppOpen-AT, Proceedings of the Eleventh International Workshop on Automatic Performance Tuning (iWAPT2016) (in conjunction with the IEEE IPDPS2016), 2016
3. Satoshi Ohshima, Takahiro Katagiri, Masaharu Matsumoto: Utilization and Expansion of ppOpen-AT for OpenACC, Proceedings of the Eleventh International Workshop on Automatic Performance Tuning (iWAPT2016) (in conjunction with the IEEE IPDPS2016), 2016
4. Kengo Nakajima, Parallel Iterative Solvers for Ill-conditioned Problems with Heterogeneous Material Properties, Procedia Computer Science 80, 1635-1645, 2016
5. Satoshi Ohshima, Takahiro Katagiri, Masaharu Matsumoto, Auto-Tuning of OpenACC using ppOpen-AT, HPC in Asia Workshop Poster Session, ISC High Performance 2016, Frankfurt, Germany, 2016
6. Naoya Nomura, Akihiro Fujii, Teruo Tanaka, Kengo Nakajima and Osni Marques, Performance Analysis of SA-AMG Method by Setting Extracted Near-Kernel Vectors, Proceedings of the 12th International Meeting on High Performance Computing for Computational Science (VECPAR 2016), Porto, Portugal, 2016
7. Takeshi Iwashita, Akihiro Ida, Takeshi Mifune, and Yasuhito Takahashi, Software Framework for Parallel BEM Analyses with H-matrices, The 17th Biennial Conf. Electromagnetic Field Computation (IEEE CEFC2016), 2016
8. Takahiro Katagiri, Satoshi Ohshima, Masaharu Matsumoto: Auto-tuning on NUMA and

Many-core Environments with an FDM Code, Proceedings of the Twelveth International Workshop on Automatic Performance Tuning (iWAPT2017) (in conjunction with the IEEE IPDPS 2017), 2017

9. Takeshi Iwashita, Akihiro Ida, Takeshi Mifune, and Yasuhito Takahashi, Software Framework for Parallel BEM Analyses with H-matrices Using MPI and OpenMP, *Procedia Computer Science*, 108, 2200-2209, 2017
10. Hideyuki Usui, Saki Kito, Masanori Nunami and Masaharu Matsumoto, Application of Block-structured Adaptive Mesh Refinement to Particle Simulation, *International Conference on Computational Science 2017 (ICCS2017)* (Zurich, Switzerland, June 12-14, 2017) (ICCS2017 Best Poster Awards)
11. Tetsuya Hoshino, Satoshi Ohshima, Toshihiro Hanawa, Kengo Nakajima, and Akihiro Ida, Pascal vs KNL: Performance Evaluation with ICCG Solver, *HPC in Asia Workshop Poster Session, ISC High Performance 2017*, Frankfurt, Germany, 2017
12. Akihiro Ida, Tadashi Ataka, Takeshi Mifune, Yasuhito Takahashi, Takeshi Iwashita and Atsushi Furuya, Application of Improved H-matrices in Micromagnetic Simulations, *Proceedings of 21st International Conference on the Computation of Electromagnetic Fields (Compumag 2017)*, Daejeon, Korea, 2017
13. Naoya Nomura, Akihiro Fujii, Teruo Tanaka, Kengo Nakajima and Osni Marques, Performance Analysis of SA-AMG Method by Setting Extracted Near-Kernel Vectors, *Lecture Notes in Computer Science (LNCS) 10150*, 52-63, 2017
14. Kengo Nakajima, and Toshihiro Hanawa, Communication-Computation Overlapping with Dynamic Loop Scheduling for Preconditioned Parallel Iterative Solvers on Multicore/Manycore Clusters, *IEEE Proceedings of 10th International Workshop on Parallel Programming Models & Systems Software for High-End Computing (P2S2 2017)* in conjunction with the 46th International Conference on Parallel Processing (ICPP 2017), Bristol, UK, 2017
15. Akihiro Ida, Tadashi Ataka, Takeshi Mifune, Yasuhito Takahashi, Takeshi Iwashita and Atsushi Furuya, Application of Improved H-matrices in Micromagnetic Simulations, *21st International Conference on the Computation of Electromagnetic Fields (Compumag 2017)*, Daejeon, Korea, 2017
16. Naoki Tominaga, Takeshi Mifune, Akihiro Ida, Yusuke Sogabe, Takeshi Iwashita, Naoyuki Amemiya, Application of hierarchical matrices to large-scale electromagnetic field analyses of coils wound with coated conductors, *Proceedings of the 25th International Conference on Magnet Technology (MT25)*, Amsterdam, the Netherlands, 2017
17. Masatoshi Kawai, Akihiro Ida and Kengo Nakajima, Hierarchical Parallelization of Multicoloring Algorithms for Block IC Preconditioners, *IEEE Proceedings of the 19<sup>th</sup> International Conference on High Performance Computing and Communications (HPCC 2017)*, Bangkok, Thailand (in press), 2017

(2) その他の著作物(総説、書籍など)

1. 岩下武史, 伊田明弘, 美船 健, 高橋康人, H行列を用いた並列境界要素解析フレームワークとその応用(特集 積分方程式の近似解法 : 高速多重極法・H 行列法とその応用), *計算工学(日本計算工学会誌)* 21-4, 3514-3517, 2016
2. 伊田明弘, 大規模科学技術計算に向けた低ランク行列近似手法(特集 積分方程式の近似解法 : 高速多重極法・H 行列法とその応用), *計算工学(日本計算工学会誌)*, 3502-3505, 2016
3. 伊田明弘, 階層構造を有する低ランク行列近似手法の数値計算とデータ構造, *シミュレーション(日本シミュレーション学会誌)* 35-3, 30-36, 2016

(3)国際学会発表及び主要な国内学会発表

① 招待講演 (国内会議 2 件、国際会議 12 件)

〈国内〉

1. 中島研吾, 岩下武史, 片桐孝洋, 松岡聡, ポストムーア時代の科学技術アプリケーションとアルゴリズム, 第 64 回応用物理学会春季学術講演会 特別シンポジウム「半導体とコンピュータ都のハーモニックイノベーション」(横浜, 2017 年 3 月 14 日)
2. 中島研吾, Oakforest-PACS の紹介と計算科学の進展, 第 30 回人工物コロキウム「大規模シミュレーションと人工物工学」, 東京大学人工物工学研究センター(柏, 千葉, 2017 年 3 月 15 日)

〈国際〉

3. \* Kengo Nakajima, Extreme Simulations on ppOpen-HPC, Challenges in Extreme Engineering Algorithms, ISC High Performance 2016 (Frankfurt, Germany, June 19-23, 2016)
4. \* Akihiro Ida, Application of Hierarchical Matrices with Adaptive Cross Approximation to Large-scale Simulations, Algorithms for Extreme Scale in Practice, ISC High Performance 2016 (Frankfurt, Germany, June 19-23, 2016)
5. Kengo Nakajima, Akihiro Ida and Takahiro Katagiri, Development of Large-Scale Scientific & Engineering Applications on Post-Peta/Exascale Systems using ppOpen-HPC, Enabling Technologies and Simulation Practices for High Performance Computational Mechanics, The 12th World Congress on Computational Mechanics (WCCM XII) (Seoul, Korea, July 24-29, 2016)
6. Kengo Nakajima, Parallel Iterative Solvers for Ill-conditioned Problems with Heterogeneous Material Properties, NCU Mathematics Seminar, National Central University (Taoyuan, Taiwan, November 3, 2016)
7. Kengo Nakajima, Parallel Iterative Solvers with Preconditioning in the Post-Moore Era, TIMS Seminar on Scientific Computing, National Taiwan University (Taipei, Taiwan, November 4, 2016)
8. Kengo Nakajima, Communication/Synchronization Avoiding/Reducing in Krylov Iterative Solvers on Multicore/Manycore Clusters -Early Case Studies on Oakforest-PACS with Intel Xeon Phi, Berkeley Lab Computing Sciences Seminar (Berkeley, CA, USA, January 20, 2017)
9. Kengo Nakajima, Parallel Preconditioning Method for Iterative Linear Solvers on Multicore/Manycore Clusters, IACM 19th International Conference on Finite Elements in Flow Problems - FEF 2017: Iterative Methods and Parallel Computing (Rome, Italy, April 4, 2017)
10. Akihiro Ida and Kengo Nakajima, ppOpen-HPC in Post-K/Post-Moore Era, First French-Japan-German Workshop on Programming and Computing for Exascale and beyond (Tokyo, Japan, April 5, 2017).
11. Kengo Nakajima, Communication-Computation Overlapping with Dynamic Loop Scheduling for Preconditioned Parallel Iterative Solvers on Multicore/Manycore Clusters (Berkeley, CA, USA, June 9, 2017)
12. \* Kengo Nakajima, Toshihiro Hanawa and Tetsuya Hoshino, Parallel Preconditioned Iterative Solvers on Manycore Architectures, MS26: Computing Bulks of Inner Eigenpairs of Large Sparse Matrices: From Applications and Algorithms to Performance and Software Engineering (2/2), PASC17 (The Platform for Advanced Scientific Computing) (Lugano, Switzerland, June 27, 2017)
13. Kengo Nakajima, Akihiro Ida, Masatoshi Kawai, and Takahiro Katagiri, Development of Large-Scale Scientific & Engineering Applications on Post-Peta/Exascale Systems using

ppOpen-HPC, APSCIT 2017 Annual Meeting (Asia Pacific Society for Computing and Information Technology (Sapporo, Hokkaido, Japan, July 29, 2017)

14. Takahiro Katagiri, Auto-tuning Fusion Applications, US-Japan JIFT Exascale Computing Workshop (Tokyo, August 7-8, 2017)

② 口頭発表 (国内会議 32 件、国際会議 12 件)

〈国内〉

1. 伊田明弘, 平石拓, 岩下武史, 階層型行列法の計算負荷予測と負荷分散法, 第 21 回計算工学講演会(日本計算工学会)(新潟, 2016 年 5 月 31 日~6 月 2 日)
2. 中島研吾, ポストムーア時代の大規模数値計算アルゴリズム, 第 21 回計算工学講演会(日本計算工学会)(新潟, 2016 年 5 月 31 日~6 月 2 日)
3. 片桐孝洋, 松本正晴, 大島聡史, ポストムーア時代に向けた階層型自動チューニング機能の性能評価, 第 21 回計算工学講演会(日本計算工学会)(新潟, 2016 年 5 月 31 日~6 月 2 日)
4. 中島研吾, ppOpen-HPC と大規模シミュレーション, 第 21 回計算工学講演会(日本計算工学会)(新潟, 2016 年 5 月 31 日~6 月 2 日)
5. 松岡聡, 天野英晴, 中島研吾, 井上弘士, 工藤知宏, 丸山直也, 田浦健次朗, 岩下武史, 片桐孝洋, 塙敏博, 遠藤敏夫, **ポストムーア時代における FLOPS から BYTES への変革**, 情報処理学会研究報告(2016-HPC-155-32), 2016 年並列/分散/協調処理に関する『松本』サマー・ワークショップ(SWoPP 松本 2016)(松本, 2016 年 8 月 8 日~10 日)
6. 片桐孝洋, 松本正晴, 大島聡史, **3 次元積層技術による高メモリバンド幅時代の自動チューニング~FDM コードを例にして~**, 情報処理学会研究報告(2016-HPC-155-38), 2016 年並列/分散/協調処理に関する『松本』サマー・ワークショップ(SWoPP 松本 2016)(松本, 2016 年 8 月 8 日~10 日)
7. 大島聡史, 伊田明弘, 河合直聡, 塙敏博, **階層型行列ベクトル積のメニーコア向け最適化**, 情報処理学会研究報告(2016-HPC-155-39), 2016 年並列/分散/協調処理に関する『松本』サマー・ワークショップ(SWoPP 松本 2016)(松本, 2016 年 8 月 8 日~10 日)
8. 塙敏博, 伊田明弘, 大島聡史, 河合直聡, FPGA を用いた階層型行列ベクトル積, 情報処理学会研究報告(2016-HPC-155-40), 2016 年並列/分散/協調処理に関する『松本』サマー・ワークショップ(SWoPP 松本 2016)(松本, 2016 年 8 月 8 日~10 日)
9. 伊田明弘, 階層型行列表現を用いた行列演算, 2016 年並列/分散/協調処理に関する『松本』サマー・ワークショップ(SWoPP 松本 2016)(松本, 2016 年 8 月 8 日~10 日)
10. 中島研吾, 非同期並列前処理手法について, 2016 年並列/分散/協調処理に関する『松本』サマー・ワークショップ(SWoPP 松本 2016)(松本, 2016 年 8 月 8 日~10 日)
11. 河合直聡, 伊田明弘, 岩下武史, 中島浩, XeonPhi 上での乗法シュワルツ-ブロック化多色順序付け GS 法を適用した SA-AMG の性能評価, 日本応用数理学会 2016 年度年会(北九州, 福岡, 2016 年 9 月 12 日~14 日)
12. 五十嵐亮, 伊田明弘, 長距離相互作用に対する 近似を用いたモンテカルロシミュレーション, 日本物理学会 2016 年秋季大会(金沢, 2016 年 9 月 13 日~16 日)
13. 塙敏博, 中島研吾, 大島聡史, 伊田明宏, 星野哲也, 田浦健次朗, データ解析・シミュレーション融合スーパーコンピュータシステム Reedbush-U の性能評価, 情報処理学会研究報告(2016-HPC-156-10), 日本情報処理学会第 156 回 HPC 研究会(小樽, 2016 年 9 月 15 日~16 日)
14. 塙敏博, 中島研吾, 大島聡史, 星野哲也, 伊田明弘, **パイプライン型共役勾配法の性能評価**, 情報処理学会研究報告(2016-HPC-157-6), 日本情報処理学会第 157 回 HPC 研究会(那覇, 2016 年 12 月 21 日~22 日)
15. 中島研吾, 大島聡史, 塙敏博, 星野哲也, 伊田明弘, **ICCG 法ソルバの Intel Xeon Phi 向け最適化**, 情報処理学会研究報告(2016-HPC-157-16), 日本情報処理学会第 157 回 HPC 研究会(那覇, 2016 年 12 月 21 日~22 日)

16. 河合直聡, 伊田明弘, 中島研吾, 悪条件問題に対する CG 法向け IC 前処理手法の改善, 情報処理学会研究報告(2017-HPC-158-9), 日本情報処理学会第 158 回 HPC 研究会(熱海, 2017年 3 月 8 日~10 日)
17. 星野哲也, 大島聡史, 埜敏博, 中島研吾, 伊田明弘, OpenACC を用いた ICCG 法ソルバの Pascal GPU における性能評価, 情報処理学会研究報告(2017-HPC-158-18), 日本情報処理学会第 158 回 HPC 研究会(熱海, 2017年 3 月 8 日~10 日)
18. 埜敏博, 中島研吾, 大島聡史, 星野哲也, 伊田明弘, Xeon Phi+OmniPath 環境における OpenMP, MPI 性能最適化, 情報処理学会研究報告(2017-HPC-158-21), 日本情報処理学会第 158 回 HPC 研究会(熱海, 2017年 3 月 8 日~10 日)
19. 五十嵐亮, 伊田明弘, 長距離相互作用に対する 近似を用いたモンテカルロシミュレーション II, 日本物理学会第 72 回年次大会(大阪, 2017 年 3 月 17 日~20 日)
20. 埜 敏博, 星野 哲也, 中島 研吾, 大島 聡史, 伊田 明宏, GPU 搭載スーパーコンピュータ Reedbush-H の性能評価, 情報処理学会研究報告(2017-HPC-159-9), 日本情報処理学会第 159 回 HPC 研究会(柏, 千葉, 2017 年 4 月 17 日)
21. 中島研吾, 藤田敬文, ポストムーア時代の並列反復法アルゴリズム, 第 22 回計算工学講演会(日本計算工学会)(大宮, 2017 年 5 月 31 日)
22. 片桐孝洋, 大島聡史, 松本正晴, ポストムーア時代における有限差分法コードの自動チューニング技法の一考察, 第 22 回計算工学講演会(日本計算工学会)(大宮, 2017 年 5 月 31 日)
23. 伊田明弘, 片桐孝洋, 階層型行列法向け低ランク行列近似手法の検討, 第 22 回計算工学講演会(日本計算工学会)(大宮, 2017 年 6 月 1 日)
24. 河合直聡, 伊田明弘, 中島研吾, ICCG 法向け並列多色順序付け法の提案, 第 22 回計算工学講演会(日本計算工学会)(大宮, 2017 年 6 月 1 日)
25. 大島聡史, 山崎市太郎, 伊田明弘, 横田理央, GPU クラスタ上における階層型行列計算の最適化, 情報処理学会研究報告(2017-HPC-160-14), 2017 年並列/分散/協調処理に関する『秋田』サマー・ワークショップ(SWoPP 秋田 2016)(秋田, 2017 年 7 月 26 日~28 日)
26. 星野 哲也, 伊田 明弘, 埜 敏博, 中島 研吾, 階層型行列法ライブラリ HACApK を用いたアプリケーションのメニーコア向け最適化, 情報処理学会研究報告(2017-HPC-160-15), 2017 年並列/分散/協調処理に関する『秋田』サマー・ワークショップ(SWoPP 秋田 2016)(秋田, 2017 年 7 月 26 日~28 日)
27. 伊田明弘, 河合直聡, 階層型行列法における行列分割法, 2017 年並列/分散/協調処理に関する『秋田』サマー・ワークショップ(SWoPP 秋田 2016)(秋田, 2017 年 7 月 26 日~28 日)
28. 中島研吾, 並列マルチグリッド前処理付き共役勾配法のメニーコアクラスタ向け最適化, 2017 年並列/分散/協調処理に関する『秋田』サマー・ワークショップ(SWoPP 秋田 2016)(秋田, 2017 年 7 月 26 日~28 日)
29. Takahiro Katagiri, Mu Yang, Weichung Wang, Akihiro Ida, Performance Evaluation of Integration of Multiple Randomized Low-Rank Singular Value Decompositions, 2017 年並列/分散/協調処理に関する『秋田』サマー・ワークショップ(SWoPP 秋田 2016)(秋田, 2017 年 7 月 26 日~28 日)
30. 櫻井隆雄, 中島研吾, 疎行列における非零要素の値を考慮したリオーダーリング手法の検討, 情報処理学会研究報告(2017-HPC-160-30), 2017 年並列/分散/協調処理に関する『秋田』サマー・ワークショップ(SWoPP 秋田 2016)(秋田, 2017 年 7 月 26 日~28 日)
31. 野村直也, 中島研吾, 藤井昭宏, 田中輝雄, 高並列環境下に向けた粗いレベルでニアカーネルベクトルを抽出および追加的に設定する手法の性能評価, 情報処理学会研究報告(2017-HPC-160-31), 2017 年並列/分散/協調処理に関する『秋田』サマー・ワークショップ(SWoPP 秋田 2016)(秋田, 2017 年 7 月 26 日~28 日)
32. 野村直也, 中島研吾, 藤井昭宏, SA-AMG 法におけるニアカーネルベクトル抽出手法の

性能評価, 日本応用数理学会 2017 年度年会(東京, 2017 年 9 月 6 日~8 日)

〈国際〉

33. Kengo Nakajima, Thread-Scalable FEM, 2016 SIAM Conference on Parallel Processing for Scientific Computing (SIAM PP16) (Paris, France, April 12-14, 2016)
34. Takeshi Iwashita, Naokazu Takemura, Akihiro Ida, Hiroshi Nakashima, ICB: IC Preconditioning with a Fill-in Strategy Considering Simd Instructions, 2016 SIAM Conference on Parallel Processing for Scientific Computing (SIAM PP16) (Paris, France, April 12-14, 2016)
35. Takahiro Katagiri, Masaharu Matsumoto, Satoshi Ohshima, Auto-Tuning of Hierarchical Computations with ppOpen-AT, 2016 SIAM Conference on Parallel Processing for Scientific Computing (SIAM PP16) (Paris, France, April 12-14, 2016)
36. Satoshi Matsuoka, Hideharu Amano, Kengo Nakajima, Koji Inoue, Tomohiro Kudoh, Naoya Maruyama, Kenjiro Taura, Takeshi Iwashita, Takahiro Katagiri, Toshihiro Hanawa, Toshio Endo. From FLOPS to BYTES: Disruptive Change in High-Performance Computing towards the Post-Moore Era, Proceedings of the ACM International Conference on Computing Frontiers (CF'16) (Como, Italy, May 16-18, 2016)
37. Masaharu Matsumoto, Development of Scientific Applications using Numerical Libraries in ppOpen-HPC, Research and Education of Computational Science (RECS2016) (Tokyo, Japan, November 29-30, 2016)
38. Takahiro Katagiri, Masaharu Matsumoto, Satoshi Ohshima, Revolution and Auto-Tuning for Matrix Computations in Post Moore Era, 2017 SIAM Conference on Computational Science and Engineering (SIAM CSE17) (Atlanta, GA, USA, February 27-March 3, 2017)
39. Kengo Nakajima, Application Development Framework for Manycore Architectures in Exascale/Post-Moore Era, 2017 SIAM Conference on Computational Science and Engineering (SIAM CSE17) (Atlanta, GA, USA, February 27-March 3, 2017)
40. Takeshi Iwashita, Shigeru Kawaguchi, Takeshi Mifune, Tetsuji Matsuo, Automatic Mapping Operator Construction for a Subspace Correction Method Applied to a Series of Linear Systems, 2017 SIAM Conference on Computational Science and Engineering (SIAM CSE17) (Atlanta, GA, USA, February 27-March 3, 2017)
41. Kengo Nakajima, Hiding communications in preconditioned Krylov iterative solvers on multicore/manycore clusters, 2017 Conference on Advanced Topics and Auto Tuning in High Performance Scientific Computing (2017 ATAT in HPSC), National Taiwan University (Taipei, Taiwan, March 10-11, 2017)
42. Satoshi Ohshima, Directives tuning directives, 2017 Conference on Advanced Topics and Auto Tuning in High Performance Scientific Computing (2017 ATAT in HPSC), National Taiwan University (Taipei, Taiwan, March 10-11, 2017)
43. Takahiro Katagiri, Impact of Auto-tuning to Many-core and NUMA Environments, 2017 Conference on Advanced Topics and Auto Tuning in High Performance Scientific Computing (2017 ATAT in HPSC), National Taiwan University (Taipei, Taiwan, March 10-11, 2017)
44. Akihiro Ida, Low Rank Approximation Methods Used in Hierarchical Matrices, 2017 Conference on Advanced Topics and Auto Tuning in High Performance Scientific Computing (2017 ATAT in HPSC), National Taiwan University (Taipei, Taiwan, March 10-11, 2017)

③ ポスター発表 (国内会議 0 件、国際会議 1 件)

〈国内〉

#### 〈国際〉

1. Ryo Igarashi, Akihiro Ida, Monte Carlo simulation using approximation to long range interactions, Research and Education of Computational Science (RECS2016) (Tokyo, Japan, November 29–30, 2016)

#### (4)知財出願

- ①国内出願 (0 件)
- ②海外出願 (0 件)
- ③その他の知的財産権

#### (5)受賞・報道等

##### ① 受賞

\* Hideyuki Usui, Saki Kito, Masanori Nunami and Masaharu Matsumoto, Application of Block-structured Adaptive Mesh Refinement to Particle Simulation, International Conference on Computational Science 2017 (ICCS2017) (Zurich, Switzerland, June 12–14, 2017) (ICCS2017 Best Poster Awards)

##### ②マスコミ(新聞・TV等)報道

##### ② その他

#### (6)成果展開事例

##### ①実用化に向けての展開

- JHPCN(学際大規模情報基盤共同利用共同研究拠点)共同研究課題等への適用
  - ポストペタスケールシステムを目指した二酸化炭素地中貯留シミュレーション技術の研究開発(山本肇博士(大成建設)他)(H25 年度~):有限体積法による二酸化炭素地中貯留シミュレーションコードにおける並列線形ソルバのハイブリッド並列化, 並列可視化機能の導入を実施している。特に疎行列格納形式として Block ELL 法を採用することによって, Fujitsu PRIMEHPC FX10(Oakleaf-FX)1,440 ノード使用時に従来 2-3 倍の計算性能が得られている。

##### ②社会還元的な展開活動

本研究の成果はインターネット(URL: <http://ppopenhpc.cc.u-tokyo.ac.jp>)で公開し, 一般に情報提供している。ソフトウェア, ドキュメントのダウンロードも可能である。下記に個別の共同研究事例, 適用事例等を示す:

- 講義・講習会での使用
  - 東京大学情報基盤センターで実施しているお試しアカウント付き講習会において教材として使用し, 普及に努めている。
  - ◇ ライブラリ利用:高性能プログラミング初級入門(2016年9月, 2017年2月, 2017年9月)(ppOpen-APPL/FDM)

## § 5 研究期間中の活動

### 5. 1 主なワークショップ、シンポジウム、アウトリーチ等の活動

年月日	名称	場所	参加人数	概要
2016年1月22日	ESSEX-II キックオフミーティング	Berlin, Germany	20	ESSEX-II のキックオフミーティング(非公開)
2016年1月25～27日	SPPEXA-II 合同ワークショップ	ミュンヘン工科大学	200	SPPEXA-II 全体ワークショップ
2016年6月20日	ESSEX-II ミーティング	Frankfurt, Germany	5	ISC 会場での進捗報告, 打合せ(非公開)
2016年6月24日	ESSEX-II ミーティング	DLR, Cologne, Germany	10	東大情報基盤センター, DLR, BUW, FAU 合同打合せ(非公開)
2016年9月6～7日	第62回お試しアカウント付き並列プログラミング講習会	東京大学情報基盤センター	20	「ライブラリ利用:科学技術計算の効率化入門」, ppOpen-HPC 講習会
2016年9月19～22日	ESSEX-II Coding Week	FAU, Erlangen, Germany	20	ESSEX-II 開発のための集中作業(非公開)
2016年11月14日	ESSEX-II ミーティング	Salt Lake City, UT	10	SC16 会場での進捗報告, 打合せ(非公開)
2017年2月28日～3月1日	第71回お試しアカウント付き並列プログラミング講習会	東京大学情報基盤センター	20	「ライブラリ利用:科学技術計算の効率化入門」, ppOpen-HPC 講習会
2017年3月20～22日	SPPEXA-II 合同ワークショップ	ミュンヘン工科大学	200	SPPEXA-II 全体ワークショップ
2017年3月20～21日	ESSEX-II ミーティング	ミュンヘン工科大学	20	ESSEX-II 進捗報告, 打合せ(非公開)
2017年4月5日	French-Japanese-German Workshop on Programming and Computing for Exascale & beyond	在日フランス大使館	100	SPPEXA-II に属する MYX, Smart-DASH, ESSEX-II を中心とした日独仏合同ワークショップ(招待者のみ)
2017年4月6日	SPPEXA Workshop Japan 2017	秋葉原	50	SPPEXA-II に属する MYX, Smart-DASH, ESSEX-II の合同ワークショップ
2017年4月7日	第27回 ASE 研究会 (Advanced Supercomputing Environment)	東京大学情報基盤センター	20	Faisal Shahzad 氏 (FAU) の講演を中心とした公開セミナー
2017年6月19日	ESSEX-II ミーティング	Frankfurt, Germany	4	ISC 会場での進捗報告, 打合せ(非公開)
2017年6月23日	ESSEX-II ミーティング	DLR, Cologne, Germany	10	東大情報基盤センター, DLR, FAU 合同打合せ(非公開)
2017年6月27日	PASC17 (The Platform for Advanced Scientific Computing)	Lugano, Switzerland	50	国際会議 PASC17 にて ESSEX-II メンバーを中心としたミニシンポジウム



			「Computing Bulks of Inner Eigenpairs of Large Sparse Matrices: From Applications and Algorithms to Performance and Software Engineering」をオーガナイズ
2017 年 9 月 12～13 日 (予 定)	第 62 回お試しアカウント 付き並列プログラミング 講習会	東京大学情 報基盤セン ター	「ライブラリ利用: 科学技術 計算の効率化入門」, ppOpen-HPC 講習会
2017 年 10 月 18 日 (予定)	SPPEXA Workshop France 2017	Versailles, France	SPPEXA-II に属する MYX, Smart-DASH, ESSEX-II の 合同ワークショップ
2018 年 3 月 7 ～ 10 日 (予 定)	18th SIAM Conference on Parallel Processing for Scientific Computing (SIAM PP18)	Waseda, Tokyo, Japan	国際会議 SIAM PP18 にて ESSEX-II メンバーを中心と したミニシンポジウム 「Performance Engineering from the node level to the extreme scale」をオーガナイ ズ(予定)