ppOpen-APPL/FVM を使用した並列有限要素法アプリケーション

中島研吾[†1 †2]　塙 敏博[†1]　大島聡史[†1 †2]　片桐孝洋[†1 †2]

ppOpen-APPL/FVM は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 の提供する非構造格子向けアプリケーション開発フレームワークである．本研究では，ppOpen-APPL/FVM による並列有限要素法アプリケーションの開発事例，係数行列生成部と線形ソルバーに注目した性能評価を Intel Xeon Phi を搭載した PC クラスタで実施した結果を紹介する．

Parallel FEM application using ppOpen-APPL/FVM

Kengo Nakajima[†1 †2]　Toshihiro Hanawa[†1]　Satoshi Ohshima[†1 †2]
Takahiro Katagiri[†1 †2]

ppOpen-APPL/FVM is an framework for application development with unstructured meshes. ppOpen-APPL/FVM is a part of ppOpen-HPC, which is an open source infrastructure for development and execution of large-scale scientific applications on post-peta-scale supercomputers with automatic tuning (AT). In this work, a parallel FEM application developed by ppOpen-APPL/FVM is overviewed, and results of performance analyses for matrix assembly and linear solvers on PC clusters with Intel Xeon Phi are demonstrated.

1. はじめに

著者等は科学技術振興機構戦略的創造研究推進事業（CREST）「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」の 1 プロジェクトとして実施されている「ppOpen-HPC : 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」〔1,2〕において，様々な科学技術計算手法の各計算プロセスのマルチコア，メニィコアアーキテクチャ向け最適化，ライブラリ化と自動チューニング手法の適用に関する研究開発を実施している．ppOpen-HPC は ppOpen-APPL（アプリケーション開発フレームワーク），ppOpen-MATH（汎用ライブラリ），ppOpen-AT（自動チューニング），ppOpen-SYS（システムソフトウェア）の 4 層から構成されている（図 1 参照）．

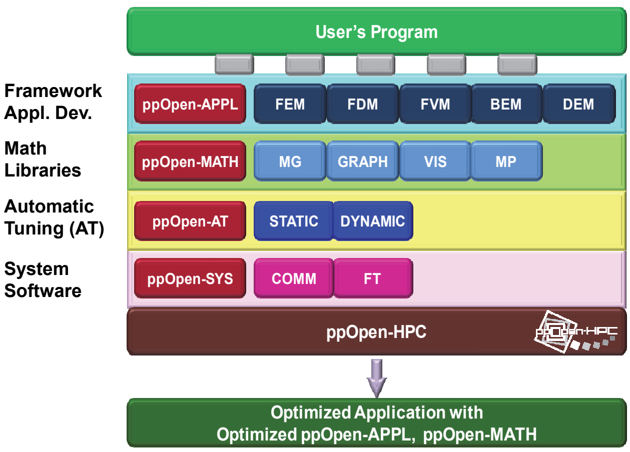

図 1　ppOpen-HPC の概要〔1,2〕

ppOpen-HPC は筑波大学と東京大学の協力のもと設置された最先端共同 HPC 基盤施設（JCAHPC）〔3〕によって導入が予定されている，メニィコアアーキテクチャに基づくポスト T2K システムをターゲットとし，利用者の新システムへの円滑な移行に資することを目的としている．ppOpen-HPC はメッセージパッシング（MPI）とプロセス内スレッド並列（OpenMP）を組み合わせたハイブリッド並列プログラミングモデルを基本としている．

ppOpen-HPC では有限要素法（FEM），有限差分法（FDM），有限体積法（FVM），境界要素法（BEM），個別要素法（DEM）など 5 種類の代表的な離散化手法に対応している．

ppOpen-APPL/FVM は有限体積法に関するアプリケーション開発フレームワークであり，非構造格子に関するインタフェース，適応格子（Adaptive Mesh Refinement，AMR）に関する機能等を提供している．三次元圧縮性粘性流体解析アプリケーション（陽解法）〔4〕を元に整備したものであり，辺に関する射影面積計算機能などを有している．三角柱，六面体，四面体等の要素をサポートしているため，有限要素法コード開発のプラットフォームとしても使用可能であり，係数行列生成機能，前処理付き反復法による線形ソルバー等の機能も開発され〔5〕，ppOpen-APPL/FVM の一部として公開される予定である．

本論文では，ppOpen-APPL/FVM の機能として整備中の：

- 係数行列生成機能（Matrix Assembly）
- 前処理付き反復法による線形ソルバー

を有限要素法による三次元定常熱伝導解析コードに適用し，Intel Xeon Phi を搭載した PC クラスタ上で性能を評価した．

†1 東京大学情報基盤センター
Information Technology Center, The University of Tokyo
†2 科学技術振興機構 CREST
CREST, Japan Science and Technology Agency

以下，アプリケーションの概要，係数行列生成機能，前処理付き反復法ソルバーの概要，性能評価，将来の展望について述べる．

2. 対象アプリケーションの概要

2.1 有限要素法プログラム HEAT-3D

本研究で対象としているのは，GeoFEM プロジェクト〔6,7,8〕で開発された並列有限要素法アプリケーションを元に整備した三次元定常熱伝導解析コード pHEAT-3D である．本コードは，一様な立方体メッシュから構成される解析モデル（図 2）を対象として，①等方性一様な熱伝導率，②各要素一様な体積発熱率，③Z=Zmax 平面における温度固定境界条件を適用している．

要素タイプは三次元一次六面体要素（tri-linear）であり，各要素 8 つの節点を有している．プログラムは全て OpenMP ディレクティヴを含む FORTRAN90 および MPI で記述されている．また，MPI，OpenMP，Hybrid（OpenMP＋MPI）の全ての環境で稼動する．pHEAT-3D はプログラム内部で自動的に並列分散メッシュを生成する機能を内蔵している．

Galerkin による有限要素法を適用しているため，三次元定常熱伝導問題では係数行列が対称正定な疎行列となることから，前処理を施した共役勾配法（Conjugate Gradient, CG）法によって連立一次方程式を解いている．

前処理手法としては点ヤコビ法（Point Jacobi）を使用しており，係数行列は CRS 形式（Compressed Row Storage）によって格納されている．

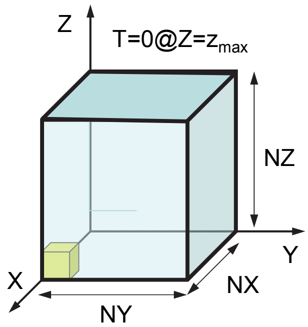

図 2　pHEAT-3D の解析対象（Cube モデル）
（NX，NY，NZ は各方向の節点数）

2.2 係数行列生成部

有限要素法では，要素毎に得られる積分方程式から導かれる密な要素行列を重ね合わせて疎な全体行列を生成する．図 3 に示すような二次元一次四角形要素（bi-linear, 双一次）では各要素の節点数が 4 であるので各節点の自由度数が 1 であれば，要素行列は 4×4 の密行列となる．

図 3 の 7 番の節点は周囲の 4 要素（2，3，5，6 番）からの寄与がある．したがって，係数行列生成のプロセスを OpenMP 等でスレッド並列化した場合，ある節点に複数の要素から同時にデータの書き込みが発生する場合がある．要素行列の重ね合わせを実施する際にはマルチカラーオーダリング等を使用してこのような同時書き込みの発生を回避する方法が広く使用されている〔9〕．

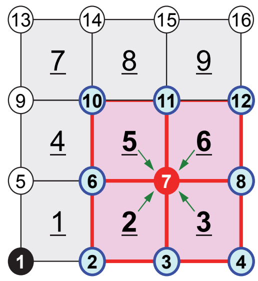

図 3　要素行列の重ね合わせによる全体行列の生成

図 4 は本研究における行列生成部の処理の概要を示すものである．三次元一次六面体要素を使用しているため，要素あたりの節点数は 8 であり，8×8 の密な要素行列が生成される．ループの構成としては一番外側が各要素に関するループ（do icel= 1, ICELTOT，ICELTOT：全要素数）である．その内側の二重ループ（do ie=1, 8，do je= 1, 8）は要素行列を生成するためのループであり，各要素の節点が 8 個あることに対応している．更にその内側にはガウスの積分公式に対応する三重ループ（do ipn/jpn/kpn =1, 8）がある．

図 4　pHEAT-3D における係数行列生成の処理

図 4 に示す処理をまとめると以下の 4 つとなる：

① 各積分点におけるヤコビアン，形状関数導関数計算
② 要素行列成分の全体行列（疎行列）におけるアドレス探索
③ ガウス数値積分，要素行列成分計算
④ 要素行列成分の全体行列への加算

図 5 は，図 4 に示した処理内容を，上記①～④を考慮して簡略化し，OpenMP によるスレッド並列化が適用されていると仮定したものである．COLORtot はマルチカラーオー

Vol.2015-HPC-151 No.24
2015/10/1

ダリングの色数であり，本ケースのような規則正しい形状の場合には 8 である．配列 col_index(color)は各色に含まれる要素数である．図 5 に示すようにオリジナル実装では，これらの処理を要素毎に実施しており，特に②～④については要素行列の各成分について個別に実施している．各ループの中で，探索，ガウス積分，全体行列への加算などの複雑な処理が繰り返し実施されるため計算効率が低くなっている可能性がある．

```
   do color= 1, COLORtot
!$OMP PARALLEL DO
   do icel= col_index(color-1)+1, col_index(color)
     <①各積分点におけるヤコビアン，形状関数導関数計算>
     do ie= 1, 8; do je= 1, 8
       <②要素行列成分の全体行列（疎行列）におけるアドレス探索>
       <③ガウス数値積分，要素行列成分計算>
       <④要素行列成分の全体行列への加算>
     enddo; enddo
   enddo
   enddo
```

図 5　pHEAT-3D オリジナル実装（Original）の概要（COLORtot：要素色数（=8），col_index(color)：各色に含まれる要素数）

著者等による先行研究〔8〕では，有限要素法による三次元弾性問題における係数行列生成部を，ブロック化に基づき，最適化するため，以下に示す 2 種類の実装（Type-A, Type-B）を提案した．ここで BLKSIZ は各ブロックに含まれる要素数，NBLK は要素ブロックの総数である．

Type-A（図 6）

- 図 4 に示した①～④の処理のうち，②，①＋③，④を分離して，3 つのループとする．
- 疎行列アドレス記憶用配列，要素行列用配列のための追加の記憶容量が必要である．

Type-B

- 図 4 に示した①～④の処理のうち，②，①＋③＋④を分離して，2 つのループとする．
- 疎行列アドレス記憶用配列のための追加の記憶容量が必要である．要素行列用配列の記憶は不要である．

```
   do color= 1, COLORtot
!$OMP PARALLEL DO
   do ib= 1, NBLK
     do blk= 1, BLKSIZE
       icel: calculated by (col_index, ib, blk)
       do ie= 1, 8; do je= 1, 8
         <②要素行列成分の全体行列（疎行列）におけるアドレス探索＋格納>
       enddo; enddo
     enddo
     do blk= 1, BLKSIZ
       icel: calculated by (col_index, ib, blk)
       <①各積分点におけるヤコビアン，形状関数導関数計算>
       do ie= 1, 8; do je= 1, 8
         <③ガウス数値積分，要素行列成分計算＋格納>
       enddo; enddo
     enddo
     do blk= 1, BLKSIZ
       icel: calculated by (col_index, ib, blk)
       do ie= 1, 8; do je= 1, 8
        <④要素行列成分の全体行列への加算>
       enddo; enddo
     enddo
   enddo
   enddo
```

図 6　Type-A 実装〔8〕の概要（COLORtot：要素色数（=8），col_index(color)：各色に含まれる要素数，NBLK：要素ブロック総数，BLKSIZ：要素ブロックサイズ，icel：要素番号）

Intel Xeon (Ivy Bridge)，Intel Xeon Phi 等のアーキテクチャでは，Type-A が高い計算性能を示した．本研究では図 4 に示すオリジナル実装と，Type-A 実装を適用した．

2.3　疎行列格納形式

疎行列計算は間接参照を含むため memory-bound なプロセスである．従って疎行列演算において，演算性能と比較してメモリ転送性能の低い昨今の計算機の性能を引き出すことは困難である．係数行列の格納形式が性能に影響することは広く知られており，様々な手法が提案されている．

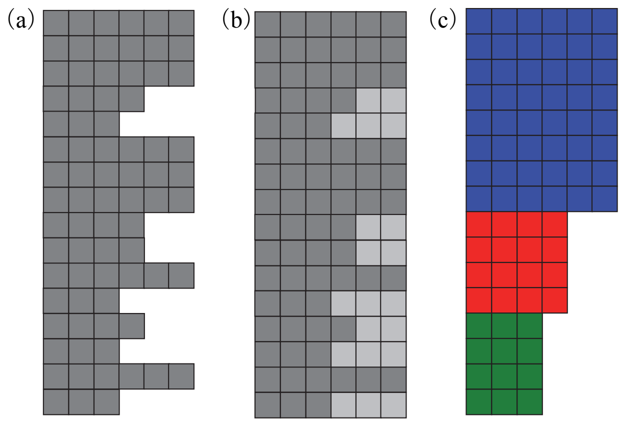

図 7　疎行列格納形式　(a) CRS (Compressed Row Storage, 非零非対角成分のみ記憶)，(b) ELL（Ellpack-Itpack，薄灰色の部分には 0 が入る)，(c) Sliced ELL〔10〕

Compressed Row Storage（CRS）形式は，図 7 (a) に示すように疎行列の非零成分のみを記憶する方法である．Ellpack-Itpack（ELL）形式は各行における非零非対角成分数を最大非零非対角成分数に固定する方法であり（図 7 (b)），実際に非零非対角成分が存在しない部分は係数=0 として計算する．CRS と比較して高いメモリアクセス効率が得られることが知られているが，計算量，必要記憶容量ともに増加する．ELL 形式を拡張し，より効率的に疎行列を記憶する手法として，Sliced ELL 形式〔10〕が提案されている（図 7 (c)）．Sliced ELL 形式は主として疎行列ベクトル積に使用されていたが，前進後退代入などデータ依存性を含むプロセスにも適用されている〔11,12〕．SELL-C-σ〔13〕は Sliced ELL を更に SIMD 向けに拡張したものである（図 8 参照）．C (Chunk Size) は SIMD

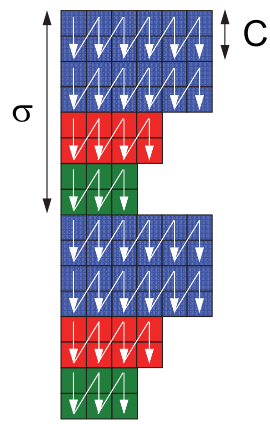

図 8　SELL-C-σ〔13〕，C：Chunk Size，σ：Sorting Scope，この図では C=2，σ=8

幅に相当し，非零非対角成分数の変化に応じて，σ（Sorting Scope）を定める．図 8 の例では，長さ 2 の Chunk が 4 つで 1 コンポーネントを構成しており，SELL-2-8 と呼ばれる．非零非対角成分が変化しない場合は SELL-C-1 と呼ばれる．

3. 計算機環境

本研究では東京大学情報基盤センターの KNSC クラスタを使用した．KNSC クラスタは 64 個の計算ノードを Infiniband 結合したものであり，各計算ノードは 2 ソケットの Intel Xeon E5（IvyBridge-EP），1 台の Intel Xeon Phi（Knights Corner）から構成されている．本研究ではそれぞれ，MIC，IvyB と呼ぶ．表 1 に MIC，IvyB の各ソケットの概要を示す．プログラムは Fortran90 で記述してあり，Intel Compiler（Ver.16）／Intel Parallel Studio XE 2016 を使用した．表 1 に計算機環境の概要を示す．

本研究では，MIC，IvyB を単独で使用した場合と，MIC，IvyB の両者を Symmetric に使用した場合〔14〕について計算を実施した．各ソケットにおいて使用したスレッド数は MIC：240，IvyB：10 である．したがって IvyB ではノード当り 20 スレッドとなる．

表 1　各計算環境（1 ソケット）の概要

略　称	MIC	IvyB
名　称	Intel Xeon Phi 5110P (Knights Corner)	Intel Xeon E5-2680 v2 (Ivy-Bridge-EP)
動作周波数（GHz）	1.053	2.80
コア数（有効スレッド数）	60 (240)	10 (20)
使用スレッド数	240	10
メモリ種別	GDDR5	DDR3
理論演算性能（GFLOPS）	1,010.9	224.0
主記憶容量（GB）	8	64
理論メモリ性能（GB/sec.）	320	59.7
キャッシュ構成	L1:32KB/core L2:512KB/core	L1:32KB/core L2:256KB/core L3:25MB/socket
コンパイルオプション	-O3 -openmp -mmic -align array64byte	-O3 -openmp -ipo -xAVX -align array32byte

4. 計算結果

4.1 1 ノードにおける計算

MIC，IvyB の 1 ノードを使用した場合について計算を実施した．計算対象は図 2 に示す Cube モデルで NX=NY=NZ=128（節点数）とした場合について検討した．したがって，要素数＝2,048,383（$=127^3$），節点数（＝自由度数）＝2,097,152（$=128^3$）である．

疎行列格納形式としては，以下の 3 種類を考慮した：

- CRS
- ELL
- SELL-C-1（C=2, 4, 8）

本研究の計算対象は規則正しい形状であり，NX=NY=NZ=128 の場合，95%以上の節点において非零非対角成分の数は一様（=26）である．そこで本研究では SELL-C-1 を採用し，C の値として 2，4，8 の場合を考慮した．ELL においては，係数行列の格納順序は CRS と同様とし〔11〕，SELL-C-1 については図 8 のように，Jagged Diagonal 式の格納順序とした〔13〕．

総スレッド数は MIC：240，IvyB：20 であり，ノード内の MPI プロセス数を 1, 2, 4 とした場合について実施した．〔11〕等の記法に従えば各ケースは表 2 のようになる．

表 2　OpenMPI/MPI ハイブリッド並列プログラミングモデルの概要

ノード内 MPI プロセス数	MIC	IvyB
1	HB 240×1	HB 20×1
2	HB 120×2	HB 10×2
4	HB 60×4	HB 5×4

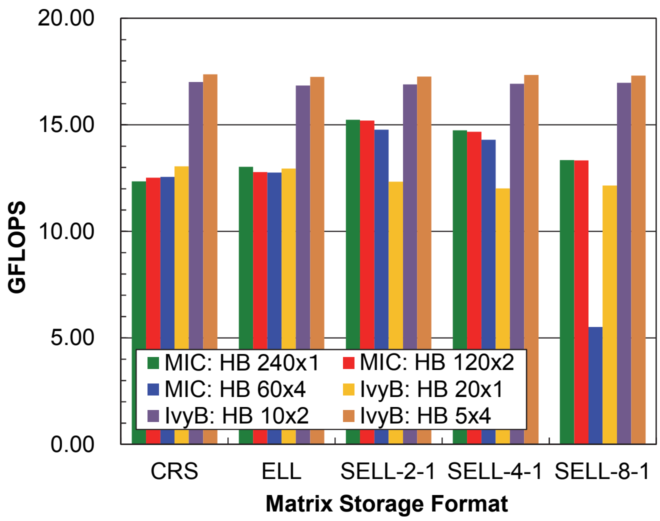

図 9　MIC 及び IvyB　1 ノードにおける計算結果，CG 法ソルバー部分の計算性能（GFLOPS 値），疎行列格納形式（CRS，ELL，SELL-2/4/8-1）・ノード内 MPI プロセス数の影響（2,097,152 自由度）

図 9 は，疎行列格納形式，各ハイブリッド並列プログラミングモデル（ノード内 MPI プロセス数）における CG 法ソルバーの計算性能（GFLOPS 値）である．HB 240×1（MIC）と HB 20×1（IvyB）を比較するとほぼ同じ性能である．IvyB については，ノード内 MPI プロセス数を増加させることによって性能が大幅に改善されるが，MIC の場合はほとんど変わらないか，SELL-C-1 においてはやや低下する傾向にあり，特に SELL-8-1 の HB 60×4 においては性能が大幅に低下している．MIC においては ELL，SELL-C-1 の採用によ

って CRS と比較して性能は向上しているが，〔11〕の場合ほど顕著でないのは，非零非対角成分数が〔11〕で扱っている 7 点ステンシルと比較して多い（26 個）ことも起因していると考えられる．SELL-2-1 の性能が最も高く，HB 240×1 の場合は CRS と比較して 23%の性能向上が得られている．IvyB については〔12〕の場合と同様，疎行列格納方法による差異は小さく，CRS の性能がやや良い．

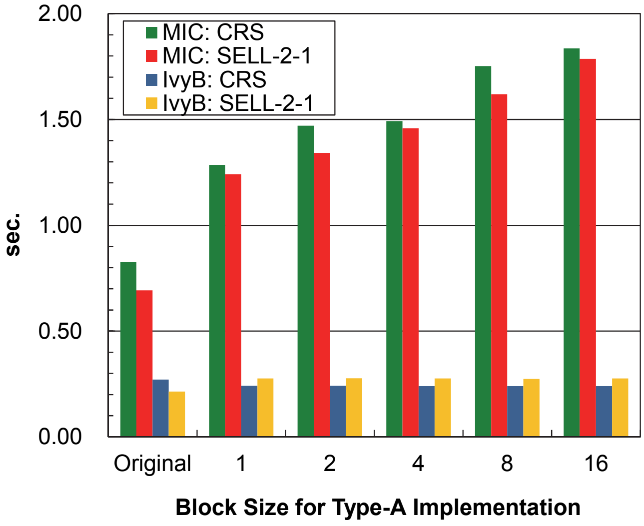

図 10　MIC 及び IvyB　1 ノードにおける計算結果，係数行列生成部（図 5，図 6）の計算時間（MIC：HB 240×1，IvyB：HB 20×1）

図 10 は係数行列生成部（図 5，図 6）の計算時間である（MIC：HB 240×1，IvyB：HB 20×1）．〔9〕では，MIC の方が IvyB と比較して高かったが，図 10 では IvyB の性能が 2 倍以上高い．〔9〕では IvyB の 1 ソケットのみ使用していたが，本研究では 2 ソケット使用していることも原因の一つであるが，pHEAT-3D においては，図 5，6 に示した実数演算部分の要素当り計算量が〔9〕の三次元弾性問題と比較して少ない（約 9 分の 1）ことも起因している．

また〔9〕では CRS 形式のみ扱ったが，疎行列格納形式によっても性能が異なっていることがわかる．また，MIC においては図 6 に示した Type-A 実装によって性能が大幅に低下しており，これも〔9〕とは異なる傾向である．

図 11，図 12 は MIC（SELL-2-1），IvyB（CRS）における計算時間の内訳である．図中では以下の 4 つのプロセスに分類している：

- Solver：共役勾配法（CG 法）計算時間
- Mat. Assembly（Matrix Assembly）：係数行列計算部分，
- Coloring：要素色分け
- Mat. Connectivity（Matrix Connectivity）：係数行列情報生成

このうち，Coloring，Mat. Connectivity の部分は並列化されておらず，1 スレッドで実行している．ノード内 MPI プロセス数が増加すると 1 スレッド当りの計算量が減るため，これらの部分の計算時間は減少している．また，1 スレッド（コア）の計算能力が MIC は IvyB に比べて小さいため，10 倍近い計算時間を要している．

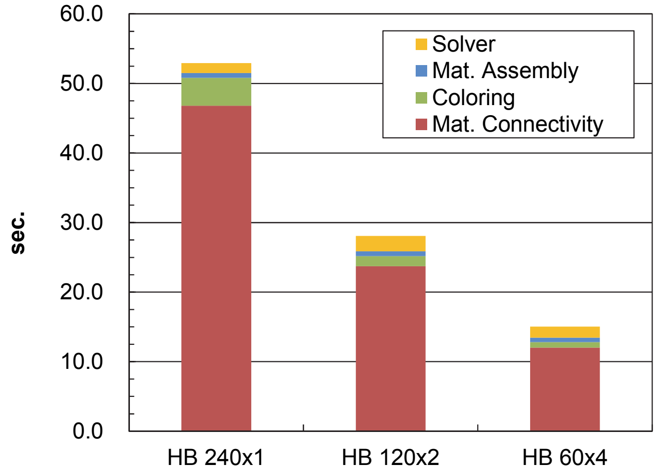

図 11　MIC 1 ノードにおける計算結果（2,097,152 自由度），SELL-2-1 の場合，Solver：共役勾配法（CG 法）計算時間，Mat. Assembly（Matrix Assembly）：係数行列計算部分，Coloring：要素色分け，Mat. Connectivity（Matrix Connectivity）：係数行列情報生成，

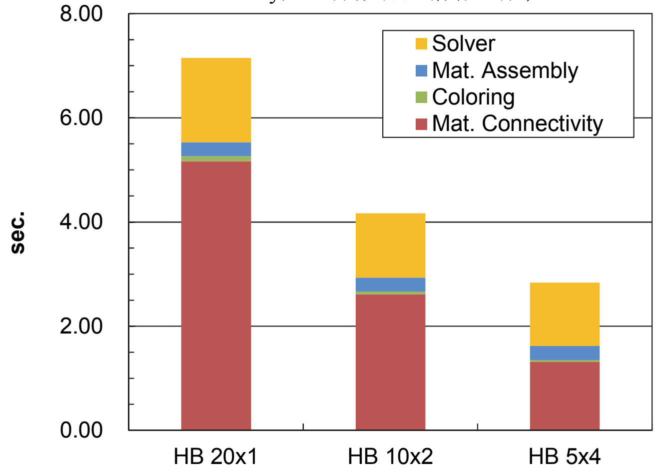

図 12　IvyB 1 ノードにおける計算結果計算結果（2,097,152 自由度），CRS の場合，Solver：共役勾配法（CG 法）計算時間，Mat. Assembly（Matrix Assembly）：係数行列計算部分，Coloring：要素色分け，Mat. Connectivity（Matrix Connectivity）：係数行列情報生成

4.2 複数ノードにおける計算

図 13 は MIC，IvyB について 1～16 ノードを使用して，CG 法ソルバーの Strong Scaling 性能を評価した結果である．計算対象は図 2 に示す Cube モデルで NX=NY=256, NZ=128（節点数）とした場合について検討した．したがって，要素数＝8,258,175，節点数（＝自由度）＝8,388,608 である．疎行列格納形式は，MIC：SELL-2-1，IvyB：CRS である．IvyB HB 20×1 における 1 ノードの計算性能を 1 として，ス

ケーラブルな性能向上に対する比を求めている．IvyB に対して MIC では 16 ノードにおける性能低下が顕著である．

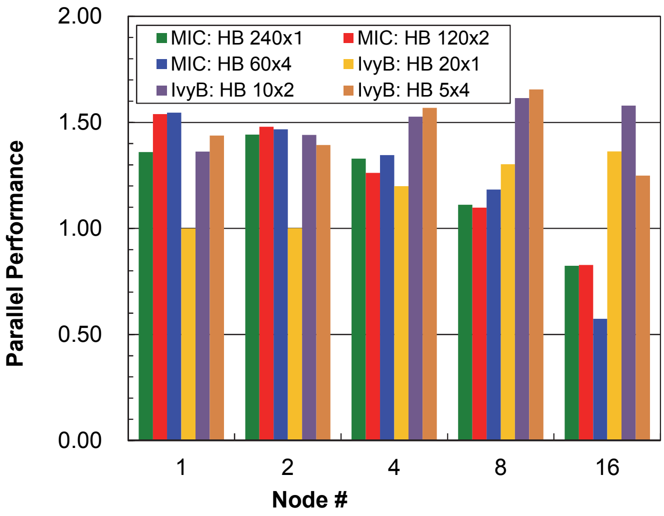

図 13　MIC（SELL-2-1）及び IvyB（CRS）における計算結果，Strong Scaling，CG 法ソルバーの計算性能，IvyB：HB 20×1 の性能を基準，8,388,608 自由度

を使用した場合について計算を実施した．計算対象は図 2 に示す Cube モデルで NX=NY=NZ=128（節点数）とした場合について検討した．したがって，要素数＝2,048,383（$=127^3$），節点数＝2,097,152（$=128^3$）である．

図 14 は図 13 に示した 4 ノードの場合に，MIC と IvyB を同時に使用し，Symmetric 実行を実施した場合の CG ソルバーの計算時間である．MIC と IvyB の各 MPI プロセスは同じサイズの問題を計算している．Symmetric 実行によって，性能が 2 倍弱改善していることがわかる．MIC（SELL-2-1），IvyB（CRS）では異なる疎行列格納形式を適用している．

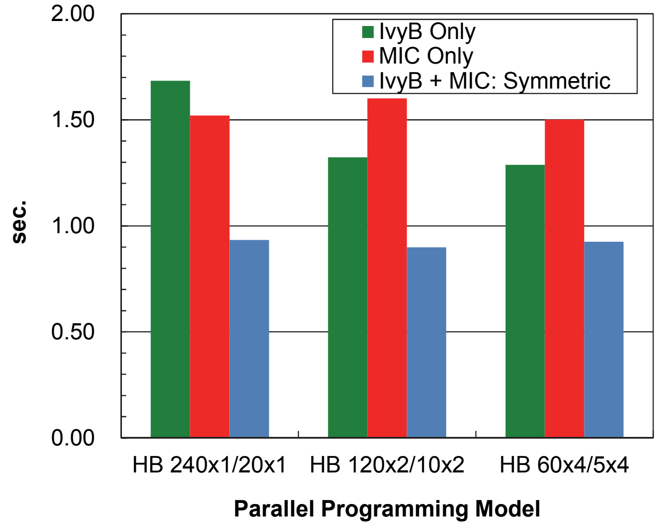

図 14　MIC（SELL-2-1），IvyB（CRS）及び Symmetric 実行における計算結果，CG 法ソルバーの計算性能，4 ノード，8,388,608 自由度

5. まとめ

本研究では，ppOpen-APPL/FVM による並列有限要素法による三次元定常熱伝導アプリケーションの開発事例，係数行列生成部と線形ソルバーに注目した性能評価を Intel Xeon Phi を搭載した PC クラスタで実施した結果を紹介した．係数行列生成部における性能については，三次元弾性力学を対象とした先行研究と異なる傾向が見られた．様々なアプリケーションに関する検討が必要である．メニィコアアーキテクチャ向けの疎行列格納方法として注目されている SELL-C-σに関する検討を実施した．MIC では一定の効果が見られたが，パラメータの影響，様々なマトリクスへの適用に関する検討が必要である．有限要素法の計算においては，疎行列情報生成，要素色分けなどのプロセスが並列化されておらず，ボトルネックとなっている．これらのプロセスの並列化に向けての検討が必要である．

参考文献

1)　ppOpen-HPC：科学技術振興機構戦略的創造研究推進事業（CREST）「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：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」，http://ppopenhpc.cc.u-tokyo.ac.jp/
2)　中島研吾，佐藤正樹，古村孝志，奥田洋司，岩下武史，阪口秀，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，情報処理学会研究報告（HPC-130-44）（2011）
3)　最先端共同 HPC 基盤施設（JCAHPC）http://jcahpc.jp/
4)　Parthasarathy, V., Kallinderis, Y., Nakajima, K., Hybrid Adaptation Method and Directional Viscous Multigrid with Prismatic / Tetrahedral Meshes, AIAA Paper 95-0670（1995）
5)　東京大学情報基盤センターお試しアカウント付き並列プログラミング講習会「ppOpen-HPC で学ぶ並列プログラミングと並列前処理付き反復法」，http://nkl.cc.u-tokyo.ac.jp/seminars/ppOpen-APPL-FVM/
6)　GeoFEM：並列有限要素法による固体地球シミュレーションプラットフォーム，http://geofem.tokyo.rist.or.jp
7)　Nakajima, K., Parallel Iterative Solvers of GeoFEM with Selective Blocking Preconditioning for Nonlinear Contact Problems on the Earth Simulator, ACM/IEEE Proceedings of SC2003, (2003)
8)　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）
9)　中島研吾，大島聡史，塙敏博，有限要素法係数行列生成プロセスのマルチコア・メニィコア環境における最適化，情報処理学会研究報告（HPC-146-22）（2014）
10)　Monakov, A., A. Lokhmotov, and A. Avetisyan, Automatically tuning sparse matrix-vector multiplication for GPU architectures, Lecture Notes in Computer Science 5952　（2010）112-125
11)　Nakajima, K., Optimization of Serial and Parallel Communications for Parallel Geometric Multigrid Method, Proceedings of IEEE ICPADS 2014（2014）25-32
12)　中島研吾，拡張型 Sliced-ELL 行列格納手法に基づくメニィコア向け疎行列ソルバー，情報処理学会研究報告（HPC-147-3）（2014）
13)　M. Kreutzer, G. Hager, G. Wellein, H. Fehske, and A. R. Bishop: A unified sparse matrix data format for efficient general sparse matrix-vector multiplication on modern processors with wide SIMD units. SIAM Journal on Scientific Computing 36-5（2014）C401–C423
14)　Jeffers, J., Reinders, J., Intel Xeon Phi Coprocessor High-Performance Programming, Morgan Kaufmann　（2013）