

電磁流体コードを用いた Xeon Phi の性能評価

深沢圭一郎^{†1, †2, †3} 岡慶太郎^{†4}

我々が開発している電磁流体 (MHD) コードは宇宙プラズマの研究に用いられている一方で、様々な計算機において実アプリケーション性能評価を行っており、それらの計算機性能を実アプリケーションで相対的に評価することが出来る。本研究では、MHD コードを用いてメニーコアプロセッサである Xeon Phi の性能評価を行い、他の計算機システムとの比較を行った。評価としては、ベクトル向け最適化、キャッシュヒット最適化を行ったコードにおいて、プロセス数とスレッド数を変化させ性能を調べた。この結果、Xeon Phi では従来の X86 系と同じくベクトル向け最適化が効き、特にスレッド数が 4、プロセス数が 60 の場合が最も性能が出る事がわかった。実効性能を考えると、最適化が行われた場合と行われない場合で、倍以上の性能差となることが分かった。

Performance measurements of Xeon Phi with Magnetohydrodynamics code

KEIICHIRO FUKAZAWA^{†1, †2, †3} KEITARO OKA^{†4}

The magnetohydrodynamics (MHD) code we have developed is used to study the space plasma and measures the performance on the various computer systems. Thus we can evaluate the performance of computer systems relatively using those performances. In this study we evaluated the newly many core co-processor Xeon Phi with MHD code and compared the performance with other computer systems (architectures). In the evaluation we use the codes which is optimized for vectorization and cache hitting with changing the number of process and thread. As the results, we found that the vector tuned code is suitable for Xeon Phi and in particular we obtained the best performance using the combination of 4 threads and 60 processes. Considering the effective performance, there is double performance difference between the optimized code and not optimized code.

1. はじめに

宇宙空間は真空と思われているが、その 99%はプラズマで満たされている。プラズマとは電離した気体のことであり、帯電している電子とイオンが分かれて存在する状態である。しばしば物質の第 4 の状態とも呼ばれている。宇宙空間、特に我々の暮らす太陽系においては太陽から太陽風と呼ばれるプラズマの風が常時吹き出しており、太陽系全体にそのプラズマが充満している。宇宙プラズマは導電率が高いため、プラズマは磁力線に沿って動きやすく、また磁力線を横切る動きを取りにくい特徴がある。そのため、太陽風プラズマは太陽の磁場を伴って超音速で吹き出しており、地球のような磁化惑星に衝突すると、その磁場を伴ったプラズマの風が惑星の固有磁場と相互作用する。その結果、惑星磁場が変形し、磁気圏という図 1 に示すような形をとる。

惑星磁気圏の太陽側は太陽風の圧力により圧縮された形をしており、反太陽側は太陽風によって引き延ばされた形をしている。図の左側から太陽風が流れ込み、磁気圏の前面には、弓形の衝撃波面、ショックフロント (bow shock)

が形成され、その内側にはマグネトシースが存在する。磁気圏は磁場構造により、内部磁気圏 (中低緯度に根ざす閉じた磁力線からなる領域) と外部磁気圏 (高緯度側に根ざす開いた磁力線からなる領域) の 2 つに分けられる。その内部磁気圏と外部磁気圏の昼側境界にあたるのが、カスプ領域である。ローブ領域は外側磁気圏で開いた磁力線領域であり、希薄なプラズマが存在している。ローブに挟まれた閉じた領域がプラズマシートと呼ばれる部分で地球の極域電離圏と磁力線を通してつながっている。磁気圏境界面 (magnetopause) と呼ばれる部分が、地球磁気圏の殻である。より詳細な紹介については、参考文献[1]などを参考にされたい。

宇宙プラズマ研究において、我々は主にこのような太陽から吹いてくる磁場を伴ったプラズマの風 (太陽風) と地球の磁場が相互作用して起こる様々な現象を研究ターゲットにしている。これらは宇宙空間で起きる現象であるため探査機を打ち上げて観測を行うが、基本的に“その場”の観測しか行えない (立体空間情報を得ることができない)。そのため、3 次元空間構造、さらにその時間発展などを調べることで宇宙プラズマ計算機シミュレーションがこの分野の理論の発展、また観測結果の理解の促進に非常に重要な役割を果たしてきている。

†1 九州大学情報基盤研究開発センター
Research Institute for Information Technology, Kyushu University

†2 九州大学国際宇宙天気科学・教育センター
International Center for Space Weather Science and Education,
Kyushu University

†3 JST, CREST

†4 九州大学大学院システム情報科学府情報知能工学専攻
Department of Advanced Information Science and Electrical Engineering,
Graduate School of Information Science and Electrical Engineering, Kyushu
University

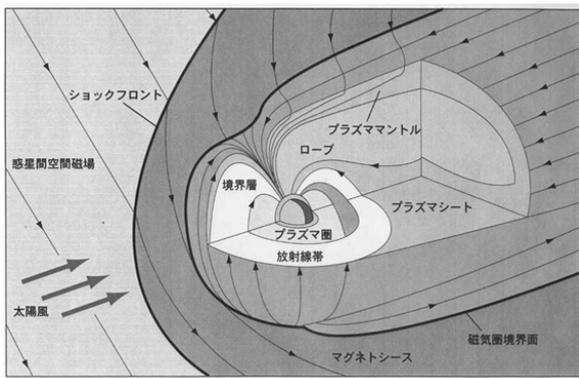


図 1 地球磁気圏構造

Figure 1 Configuration of Terrestrial magnetosphere

このようなプラズマの振る舞いを記述する方程式として Vlasov-Maxwell 方程式がある。これは、無衝突 Boltzmann 方程式と Maxwell 方程式から成る方程式で、宇宙プラズマの振る舞いを正確に記述できる。しかしながら、Vlasov 方程式は多くの成分からなる非線形方程式であり、計算機システムを用いても解くことが非常に難しい。そこで、Vlasov 方程式のモーメントをとることで求められる電磁流体力学 (MHD) 方程式が、グローバルなプラズマ構造を調べるときには使用されている。

この方程式を利用した MHD シミュレーションコードは 1980 年代に開発が始まり、現在でも様々な研究者が改良を行い、宇宙プラズマ研究に利用されているコードである。特に我々のコードは数値計算コア部分が大きく変わらず、長く利用されており、過去の計算機システムとの比較や、計算機システムの性能を確かめることに適したコードである。そこで本研究では、近年発表されたメニーコアプロセッサである Xeon Phi の性能評価を行う。

Xeon Phi は 2012 年 6 月に発表された X86 アーキテクチ

表 1 Xeon Phi サーバの諸元
 Table 1 System of Xeon Phi server

MIC	Architecture	Xeon Phi 5110P (Knights Corner)
	Number of core	60 cores
	Frequency	1.053 GHz
	Cache	L2: 30 MB/CPU
	Rmax	1.01 TFlops
	Memory size	8 GB (GDDR5)
	Bandwidth	320 GB/s
	B/F	0.32
CPU	Architecture	Xeon E5 2643 3.3 GHz
	Memory size	64 GB
System	Number of nodes	2
	MIC per node	1

ャを持つメニーコアプロセッサであり、GPGPU と同様に PCI Express スロットに接続される。本研究で利用した計算機システムの諸元を表 1 に示す。Xeon Phi は 512 ビットの SIMD 命令セットを持ち、FMA (Fused Multiply-Add) 命令もサポートするため、1 サイクル毎に単精度で 32、倍精度で 16 の浮動小数点演算が可能である。また、コアあたり 4 つのスレッドを持つ。

本研究報告の構成は以下の通りである。第 2 章では、プラズマの挙動を記述する電磁流体方程式について説明し、第 3 章では数値計算手法、並列化手法などを簡単に説明する。第 4 章で MHD コードを使用した Xeon Phi の性能評価結果を述べて、最後に研究のまとめをする。

2. MHD 方程式

宇宙プラズマの密度は非常に低いため、その平均自由行程が非常に長くなる。例えば、太陽プラズマの平均自由行程は 1 天文単位 (太陽と地球の距離) にも達する。そのため宇宙プラズマは基本的に衝突が無いと見なされる。その無衝突プラズマの振る舞いは以下の Vlasov (無衝突 Boltzmann) 方程式によって記述される。

$$\frac{\partial f_s}{\partial t} + \vec{v} \cdot \frac{\partial f_s}{\partial \vec{r}} + \frac{q_s}{m_s} (\vec{E} + \vec{v} \times \vec{B}) \cdot \frac{\partial f_s}{\partial \vec{v}} = 0 \quad (1)$$

ここで \vec{E} , \vec{B} , \vec{r} と \vec{v} はそれぞれ電場、磁場、距離、速度を表す。また、 $f_s(\vec{r}, \vec{v}_s, t)$ は位置-速度位相空間における分布関数であり、 s はイオンや電子など種類を示す。 q_s は電荷を m_s は質量を表す。MHD 方程式は(1)式、Vlasov 方程式の 0 次、1 次、2 次のモーメントをとり、運動論的效果を無視することで得られ、以下のようになる。

$$\begin{aligned} \frac{\partial \rho}{\partial t} &= -\nabla \cdot (\mathbf{v}\rho) \\ \frac{\partial \mathbf{v}}{\partial t} &= -(\mathbf{v} \cdot \nabla)\mathbf{v} - \frac{1}{\rho} \nabla p + \frac{1}{\rho} \mathbf{J} \times \mathbf{B} \\ \frac{\partial p}{\partial t} &= -(\mathbf{v} \cdot \nabla)p - \gamma p \nabla \cdot \mathbf{v} \\ \frac{\partial \mathbf{B}}{\partial t} &= \nabla \times (\mathbf{v} \times \mathbf{B}) \end{aligned} \quad (2)$$

上から、連続の式、運動方程式、圧力変化の式 (エネルギーの式)、最後が磁場の誘導方程式となる[1]。簡単に言えば、電磁場を考慮した流体力学方程式と呼べる。詳しい導出方法は参考文献を参照されたい[2]。

3. 数値モデル

3.1 シミュレーションモデル

MHD 方程式を解く数値計算法としては、Ogino らによって開発された Modified Leap Frog 法[3, 4]を使用する。これは最初の 1 回を two step Lax-Wendroff 法で解き、続く ($l - 1$) 回を Leap Frog 法で解き、その一連の手続きを繰り返す。

返す. 図2に Modified Leap Frog 法の計算スキームを示す. l の値は数値的に安定性の範囲で大きい方が望ましいので, 2次精度の中心空間差分を採用するとき, 数値精度の線形計算と予備的シミュレーションから $l=8$ に選んでいる. この手法を用いた計算で, 今まで様々なシミュレーションを行ってきたこともあり, 同様の手法をもちいることで, 過去の結果と比較できる利点がある[5].

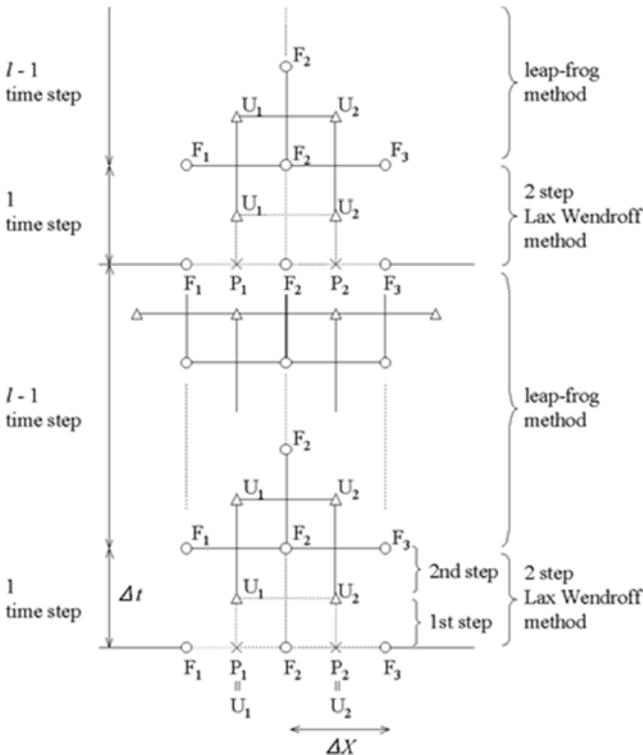


図2 Modified Leap Frog 法の計算スキーム
 Figure 2 Scheme of Modified Leap Frog method

3.2 並列化モデル

並列化には MPI と Open MP を使用する. プロセス並列化手法としては 3次元空間を分割する領域分割法を用いる

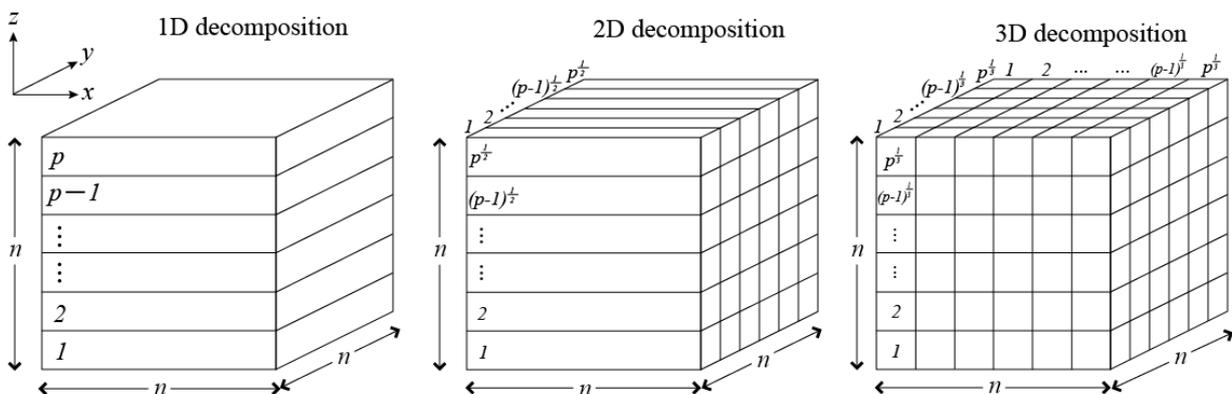


図3 1次元, 2次元, 3次元領域分割の模式図

Figure 3 Concept of domain decomposition in 1D, 2D and 3D

[5]. 領域分割には, 図3に示すように, 1次元, 2次元, 3次元分割が考えられ, 本性能評価ではこれらすべての評価を行う. 領域分割の次元数により, 計算ループのベクトル長が変わるため, それぞれの性能評価を行った.

一般的にスカラ機で性能を出すにはキャッシュの有効活用が重要である. 基本的な動作としてはデータアクセス時に, その前後含めて数 KB のデータをキャッシュに格納する. キャッシュの量や, 一度にキャッシュに格納するデータ量は CPU アーキテクチャ毎に変わるので, 最高のパフォーマンスを出すにはそれぞれの調整が必要である. MHD シミュレーションにおいては, 物理変数がプラズマ密度, 速度3成分, 圧力, 磁場3成分の計8変数となる. そのため, 配列を $u(i, j, k, m)$ (これを Type A とする) と定義し, $m = 8$ としている. 数値計算時に同じ場所の物理変数を何度も使うことになるため, 一般に $u(m, i, j, k)$ (これを Type B とする) と定義した方がキャッシュヒット率は上がることがわかっている[6]. そのため, 本性能評価においてもこの配列定義を使った性能評価も行う.

4. MHD コードの性能評価結果

今回の性能評価では 10 MB/コアの配列を計算するが, MHD 方程式を Modified Leap Frog 法で解くための作業配列として 50MB/コアを追加で使用し, 最大で 6GB のメモリを利用する. 今回利用した Xeon Phi 5110P は 8 GB のメモリを持つが, 一部は OS にも利用されているため, すべてを利用することはできない. 惑星磁気圏を解く MHD シミュレーションでは, weak scaling が重要なため, コア当たりのメモリサイズは不変とした. プログラム言語は Fortran を利用している. また流体の差分計算が主であるため, 並列化に伴う通信は袖領域の通信が支配的である. まず, MPI を用いてプロセス並列のみ (Flat MPI) で計算を行い, OpenMP と MPI を利用したハイブリッド並列については3次元領域分割のみを用いた評価を行った.

Xeon Phi を利用するには二通りの方法がある. 一つは

Native 実行と呼ばれる Xeon Phi を独立した計算サーバとして扱い、アプリケーションを Xeon Phi 上でのみ実行させる方法であり、もう一つは Offload 実行と呼ばれ、アプリケーションの一部の実行を Xeon Phi で行い、その他は CPU で実行するヘテロな実行方法である。Native 実行では X86 系システムで動作するプログラムであれば、ソースの改変を行わずに、Xeon Phi 上で動作可能である。Offload 実行の場合は、ソースに Xeon Phi が担当する計算部分を示すディレクティブを追加する必要がある。本研究では Xeon Phi 自体の性能を調べるために、Native 実行を用いて性能測定を行った。

今回利用した Xeon Phi サーバでは、Fortran コンパイラとして、Intel Fortran Compiler 13.1.3 20130607 を利用した。利用したコンパイラオプションは以下の通りである。

```
-mmic -openmp -O3 -ipo -fp-model fast=1 -no-prec-div
-no-prec-sqrt -unroll -opt-prefetch -opt-report-phase=all
-opt-threads-per-core
```

オプションの詳細は参考文献[7]を参照されたい。なお Flat MPI 利用時は `-openmp` を外してコンパイルしている。

4.1 Xeon Phi の性能評価

図 4 に MHD コードの Xeon Phi における実効性能を載せる。横軸がプロセス数、縦軸が実効性能を示す。Xeon Phi はコアあたり 4 スレッド利用できるため、最大プロセス数は 240 となっている。図 4 より Xeon Phi では 240 プロセス利用時に 2 次元領域分割、3 次元領域分割 Type A の性能がもっとも高くなり (29.15 GFlops)、キャッシュヒットを考慮した 3 次元領域分割 Type B が最も性能が出ないことが分かる。後述するが、これは今までの X86 系の傾向と同じ結果である[5]。

また、120 プロセスまではどの領域分割法でも大きく性能の差は出ていないが、240 プロセスになると差が大きい。特にキャッシュヒットを考慮した 3 次元領域分割 Type B で

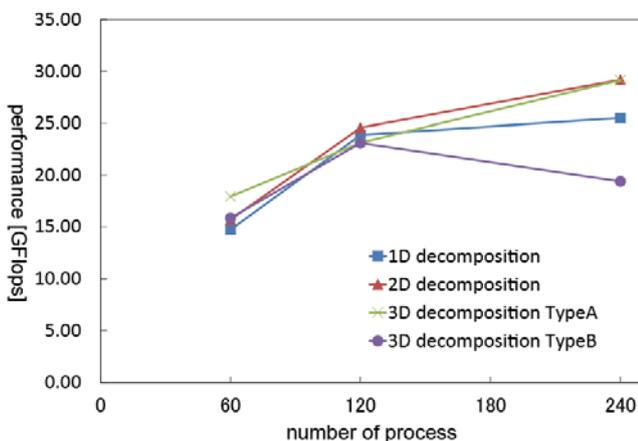


図 4 Xeon Phi における MHD コードの実効性能
 Figure 4 Performance of MHD code on Xeon Phi

は 120 プロセスに比べて、240 プロセス利用した方が性能の劣化が見える。このような振る舞いは今までの X86 系計算機システムでは見えていない[5]。240 プロセスでの性能は 3 次元領域分割 Type A だけは 60 プロセスと比べて、ほぼ線形に性能向上が見える。

次に MPI プロセス並列と OpenMP のスレッド並列を利用した場合の性能を調べるため、3 次元領域分割を用いて性能評価を行った。図 5 に 3 次元領域分割 Type A を用いて、プロセス数、スレッド数を変化させた時の MHD コードの性能を示す。横軸にプロセス数とスレッド数の組み合わせ、縦軸が実効性能を示す。まず、スレッド並列を用いることで、Flat MPI よりも性能が上がっている (Flat MPI では最大 29.15 GFlops)。最も性能が良かった組み合わせは 60 プロセス、4 スレッドで 48.97 GFlops を達成した。また 30 プロセス、8 スレッドの場合も 48.10 GFlops となっている。

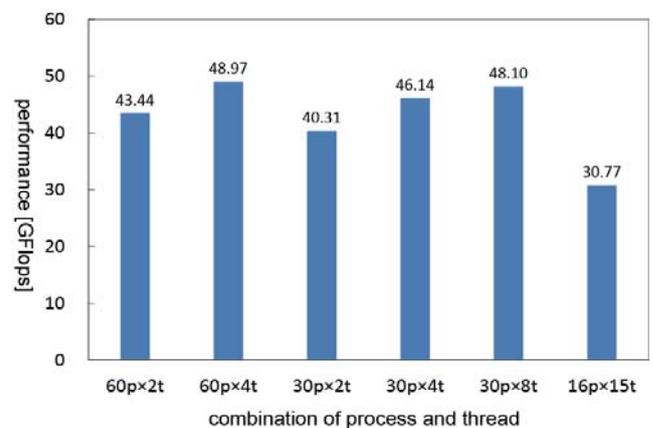


図 5 プロセス数、スレッド数を変えた場合の 3 次元領域分割 Type A における MHD コードの性能
 Figure 5 Performance of MHD code with 3D decomposition Type A in changing the number of process and thread

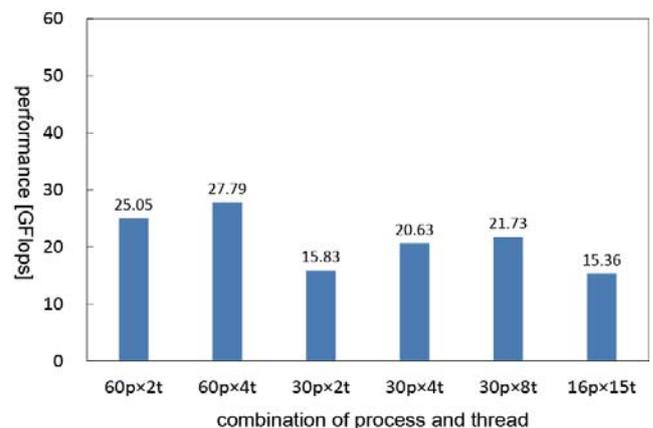


図 6 プロセス数、スレッド数を変えた場合の 3 次元領域分割 Type B における MHD コードの性能
 Figure 6 Performance of MHD code with 3D decomposition Type B in changing the number of process and thread

表 2 様々な計算機システムにおける性能の傾向[5, 6]

Table 2 Performance trend of various computer systems [5, 6]

	Core/CPU	Rpeak [TFlops]	Rmax [TFlops]	Rpeak /CPU [Gflops]	Efficiency [%]	Suitable domain decomposition	CPU architecture
SX-9	64/64	2.19	6.55	34.2	33	2D	Vector
HA8000	8192/1024	10.04	75.37	9.8	13	3D_A	Opteron (Barcelona)
HX600	1024/256	2.17	10.24	8.5	21	3D_A	Opteron (Shanghai)
XE6	8192/512	14.16	81.92	27.7	17	1D or 2D	Opteron (Interlagos)
RX200S6	864/144	3.51	10.13	24.4	35	3D_A	Xeon (Westmere)
RX200S3	1536/768	2.54	18.43	3.3	14	3D_A	Xeon (Woodcrest)
CX400	23616/2952	104.23	510.11	35.3	20	3D_A	Xeon (Sandy Bridge)
FX1	1024/256	2.08	10.24	8.1	21	3D_B	SPARC64VII
FX10	76800/4800	234.59	1135.41	48.9	21	3D_B	SPARC64 IXfx
K	262144/32768	914.12	4194.30	27.9	22	3D_B	SPARC64 VIIIfx
SR16000/L2	1344/672	5.38	25.27	8.0	21	3D_B	POWER6
Xeon Phi 5110P	60/1	0.049	1.01	49.0	5	3D_A	Knights Corner

30 プロセス、4 スレッドの場合も 46.14 GFlops と性能がある程度出ているが、240 スレッドをすべて利用した場合が最も性能が良い。120 スレッド利用する場合は 60 プロセス、2 スレッドより 30 プロセス、4 スレッドの方が良い性能となっている。一方で 16 プロセス、15 スレッドでは他の組み合わせに比べて明らかに性能が悪い。この結果と Flat MPI の結果からハイブリッド並列が Xeon Phi では効果的だが、その組み合わせには注意をすることがあることがわかる。

さらに性能が悪い場合にハイブリッド並列を行うとどのような振る舞いをするか調べるために、3次元領域分割 Type B を用いて、プロセス数とスレッド数を変化させて性能評価を行った。図 6 にその結果を示す。書式は図 5 と同じである。全体的に 3次元領域分割 Type A に比べて性能が悪いが、大まかな振る舞いは Type A に似ており、60 プロセス、4 スレッド利用時が最も良い性能となり、27.79 GFlops となっている。Flat MPI 時との性能向上率は 3次元領域分割 Type A と Type B で大きく変わらず約 1.5 倍の性能向上となっている。一方で Flat MPI で 240 プロセス時の実効性能が 19.41 GFlops のため、30 プロセス利用した性能と 16 プロセス、15 スレッド利用時の性能は Flat MPI 時とほぼ変わらないか、それより悪く、3次元領域分割 Type A の場合と異なっている。

4.2 他システムとの比較

今回の結果と今まで性能評価を行ってきた近年の計算機システムを比較するために、最大実行性能、CPU 当たりの実効性能や最適な領域分割手法を表 2 にまとめた[5, 6]。ベクトル機はベクトル長が長く取れる 2次元領域分割で性能が出ており、RISC プロセッサである POWER 系と SPARC

系ではキャッシュヒットを考慮した 3次元領域分割 Type B で性能が出ている。X86 系である Xeon 系、Opteron 系ではあまりキャッシュチューニングは効果が無く、ベクトル的な領域分割が最適という結果になっている。

一般的にベクトル機はメモリバンド幅が大きく、B/F (Bandwidth / Flops) 値が大きいため、実行効率が高いという傾向だったが、SX-9 では実効効率が下がっており、33% となっている。近年スカラチップは実行効率が上がっており、Westmere 世代の Xeon ではベクトルチップと同様の実行効率を達成している。また、AVX により SIMD が倍になり理論性能が大きく上がった Sandy Bridge 世代 Xeon においても 20% 程度の実行効率を達成している。SPARC 系では、FX10 や京で FX1 と同様の実効効率を達成しており、CPU あたりでは Xeon 系に勝る性能を達成している。

今回性能を評価した Xeon Phi では実効効率が 5% となっており、Xeon 系や SPARC 系の CPU から見ると性能が悪い。一方で CPU 当たり (Xeon Phi はコプロセッサ当たり) の性能を比べると、Xeon Phi は FX10 とほぼ同じ性能を持っている。また CX400 と比べると、約 1.4 倍の CPU 当たりの性能を持っており、単純比較は難しいが、コストまで考えても Xeon Phi の性能が高いと想定される。Xeon Phi に対しては MHD コードをまだ詳細に最適化を行っておらず、今後さらに性能が上がるようであれば、実アプリケーションの Xeon Phi 利用が広がる可能性が高い。

5. まとめ

宇宙プラズマの研究で使われている MHD コードを利用して、新しいコプロセッサである Xeon Phi の性能評価を行った。Flat MPI を利用した場合、3次元領域分割 Type A が最も性能が良く、同じアーキテクチャである Xeon や

Opteron と同じ結果であった。この 3 次元領域分割において OpenMP を利用して Hybrid MPI を用いた場合、60 プロセス、4 スレッドの場合が最も性能が良く、Flat MPI 時の約 1.4 倍の性能 (48.97 GFlops) を得ることができた Hybrid MPI を利用時には、同じ 240 スレッド利用でも性能が変わり、プロセス数とスレッド数の組み合わせに注意する必要があることが分かった。

今までに性能測定を行った別のシステムと比べると Xeon Phi は今回の最大実効効率が 4.9%であり、CPU と比べると 1/4 以下であったが、CPU 当たりの性能で比べると FX10 に搭載されている SPARC64 IXfx とほぼ同じ性能ということが分かった。本研究では詳細な最適化が行われていないが、Sandy Bridge 世代の Xeon より CPU 当たりの性能は Xeon Phi が高いという結果になった。今後詳細な最適化を行い、加えて複数ノードの評価を行っていく予定である。

謝辞 本研究の計算結果は九州大学情報基盤研究開発センターの計算機システムを利用して得られた。

参考文献

- 1) Chang, C. L. and Lee, R. C. T.: Symbolic Logic and Mechanical Theorem Proving, Academic Press, New York (1973).
- 2) R. O. Dendy, 『Plasma Dynamics』, Oxford University Press, 1990.
- 3) T. Ogino, R. J. Walker, M. Ashour-Abdalla, A global magnetohydrodynamic simulation of the magnetopause when the interplanetary magnetic field is northward, IEEE Trans. Plasma Sci.20, 817.828, 1992.
- 4) Fukazawa, K., T. Ogino, and R.J. Walker, "The Configuration and Dynamics of the Jovian Magnetosphere", J. Geophys. Res., 111, A10207, 2006.
- 5) Fukazawa, K., T. Umeda, T. Miyoshi, N. Terada, Y. Matsumoto and T. Ogino, "Performance measurement of magneto-hydro-dynamic code for space plasma on the various scalar type supercomputer systems", submitted to IEEE Trans. Plasma Sci., 2009.
- 6) Fukazawa, K., T. Nanri and T. Umeda, "Performance Measurements of MHD Simulation for Planetary Magnetosphere on Peta-Scale Computer FX10", International Conference on Parallel Computing 2013 (ParCo2013), accepted.
- 7) Intel Quick Reference Guide to Optimization with Intel Compilers Version 13.