Vol.2012-HPC-136 No.5
2012/10/3

異なるスカラアーキテクチャ（x86，SPARC64）の電磁流体コードによる性能評価

深沢圭一郎[†1, †2, †3]　南里豪志[†1, †3]　高見利也[†1, †3, †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研究に用いられている一方で，様々な計算機上で実アプリケーション性能評価を行っているため，様々な計算機の性能を実アプリケーションで相対的に評価ができる．本研究では九州大学の新しい計算機システムである Fujitsu FX10 と CX400 の性能評価を電磁流体コードを用いて行う.並列化として，4 つの手法，1 次元，2 次元，3 次元領域分割とキャッシュチューニングを行った 3 次元領域分割を用いた．この結果，FX10 では，一般にスカラ型計算機において有効であったキャッシュチューニングを施した 3 次元領域分割時に最高の性能が達成され，CX400 ではベクトル向けの 2 次元領域分割が最適であることがわかった．最終的にこれらの評価の結果，MHD コードを用い，FX10 上で 21%以上，CX400 上で 20%以上の実行効率を達成した．

Performance measurements of different scalar architectures with Magnetohydrodynamics code

KEIICHIRO FUKAZAWA[†1]　TAKESHI NANRI[†1]
TOSHIYA TAKAMI[†1]

The magnetohydrodynamics (MHD) code we have developed is used to study the space plasma and measure the performance on the various computer systems thus it can evaluate the performance of computer systems relatively. In this study we have evaluated the latest computer systems of Kyushu University, which are Fujitsu FX10 and CX400 with three-dimensional MHD code. For parallelization of the MHD code, we use four different methods, i.e., regular 1D, 2D, 3D domain decomposition methods and a cache-tuned 3D domain decomposition method. We found that the 3D decomposition with the cache-tuned of the MHD model is suitable for FX10, while the 1D decomposition method is effective to the CX400. As the results of these measurements, we achieved a performance efficiency of more than 20% on the both systems.

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のできる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 九州先端科学技術研究所
Institute of Systems, Information Technologies and Nanotechnologies

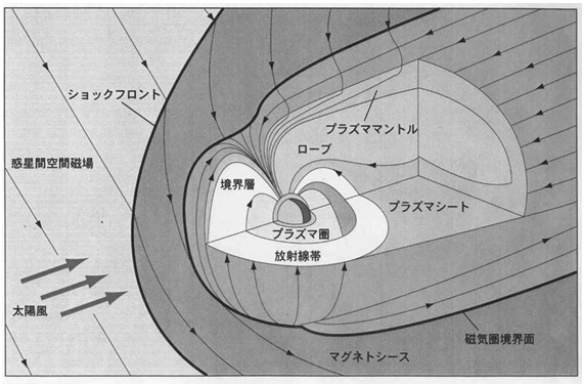

図 1　磁気圏構造
Figure 1　Configuration of magnetosphere

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，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長い間利用されているコードであり，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，九州大学に新しく導入された Fujitsu PRIMEHPC FX10 と PRIMERGY CX400 両計算機システムの性能評価を行う．

表 1　九州大学 FX10 の諸元
Table 1　System of FX10 at Kyushu University

CPU	Architecture	16 cores SPARC64 IXfx
	Frequency	1.848 GHz (236.544 GFlops)
	Cache	L1: 64KB/core L2: 12MB/CPU
Memory	Band width	85GB/s　/CPU (=node)
B/F	85/236.544	0.36
Node	Number of CPUs	1
	Memory size	32GB
System	Number of nodes	768 (12,288 cores)
	Rmax	181.6Tflops
	Node comm.	Tofu Interconnect (5GB/s)

FX10 は「京」とバイナリ互換である SPARC64 IX fx を搭載したシステムであり，九州大学に導入されたシステムの諸元を表 1 に示す．一方 CX400 は Sandy Bridge アーキテクチャの Xeon E5 を搭載した汎用的な PC クラスタタイプのシステムである．システム構成は表 2 の通りである．

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 FX10，CX400 の性能評価結果を述べて，最後に研究のまとめをする．

表 2　九州大学 CX400 の諸元
Table 2　System of CX400 at Kyushu University

CPU	Architecture	8 cores Sandy Bridge Xeon E5
	Frequency	2.7GHz (172.8GFlops)
	Cache	L2: 256KB/core L3: 20MB/CPU
Memory	Band width	51.2GB/s　/CPU
B/F	51.2/172.8	0.30
Node	Number of CPUs	2
	Memory size	128GB
System	Number of nodes	1476 (23616 cores)
	Rmax	510.1 TFlops
	Node comm.	InfiniBand FDR (6.78GB/s)

2. MHD 方程式

宇宙プラズマの密度はとても低いために，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．

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

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，運動論的効果を無視することで得られ，以下のようになる．

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

上から，連続の式，運動方程式，圧力変化の式（エネルギ

一の式)，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．l の値は数値的に安定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].

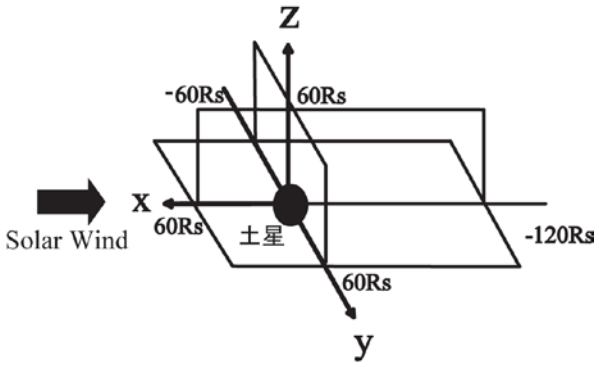

図 1　シミュレーション座標系
Figure 1　System of simulation

本シミュレーションでは，GSM (geocentric solar magnetospheric coordinates)と呼ばれる直行座標系を用いており，太陽方向を x 軸の正，夕方向を y 軸の正，北向きを z 軸の正にとっている．この座標系を図 2 に表す. そして，x 軸の正の方向から太陽風を流し続ける．

3.2 並列化モデル

並列化には MPI を使用する．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[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．分散メモリ型の並列計算機を用いた並列計算では，3 次元配列に対して領域分割を用いるのが通常である．3 次元モデルの場合，領域分割の次元を 1 次元, 2 次元, 3 次元に選ぶことができる．その場合の計算時間(T_{S1}, T_{S2}, T_{S3})と通信時間(T_{C1}, T_{C2}, T_{C3}) は大まかに次の様に見積もることができる．

i) 1 次元領域分割

$$T_{S1}=\frac{k_1 n^3}{p},\quad T_{C1}=k_2 n^2(p-1) \tag{3}$$

ii) 2 次元領域分割

$$T_{S2}=\frac{k_1 n^3}{p},\quad T_{C2}=2k_2 n^2(p^{\frac{1}{2}}-1) \tag{4}$$

iii) 3 次元領域分割

$$T_{S3}=\frac{k_1 n^3}{p},\quad T_{C3}=3k_2 n^2(p^{\frac{1}{3}}-1) \tag{5}$$

ここに，k_1 と k_2 は一定の係数，n は 3 次元配列における 1 方向の変数量, p は総並列数である．ここでは簡単のため，領域分割は図 3 に示すように x, y, z 方向に同じ数（n）の配列を使用し，各次元を並列化する場合は等しい並列化数を設定している．計算時間と通信時間の和が並列計算に要する時間と考えられる．式より明らかに計算時間 T_{S1} T_{S2} T_{S3} は並列数 p に反比例して短くなるが，通信時間 T_{C1}, T_{C2}, T_{C3} は p の増加に伴って長くなる．しかし，その通信時間の長くなる様子は，T_{C1}, T_{C2}, T_{C3} によって大きく異なる．即ち，3 次元領域分割が最も通信時間を短くでき，また，1 次元と 2 次元領域分割の間でも通信時間の差は大きくなることが理解できる．ただし，この比較では簡単のため，通信時間を決める係数 k_2 が同じであると仮定した．これは通信部分のプログラムの工夫によりある程度小さくすることが可能である．こうして，スカラ並列機では 3 次元領域分割が，一方ベクトル並列機では，1 つの次元方向はベクトル化に利用する必要があるために 2 次元領域分割が最も効率的であろうと予想できる．

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 $u(i, j, k, m)$と定義し（Type A），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ので，一般に $u(m, i, j, k)$と定義した方がキャッシュヒット率は上がると考えられる（Type B）．そのため，本性能評価においてもこの配列定義を使った評価も行う．

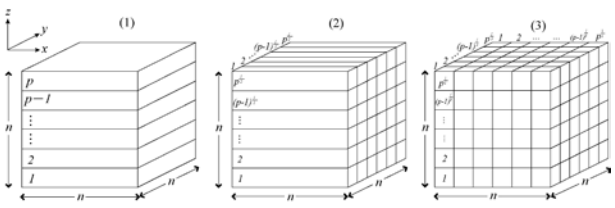

図 3　1 次元，2 次元，3 次元領域分割の模式図
Figure 3　Concept of domain decomposition in 1D, 2D and 3D

4. MHD コードの性能評価結果

今回の性能評価では 64MB/コア（1GB/ノード）サイズ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 192MB/コア（3GB/ノード）を追加で使用する．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になるため，並列化に伴う通信は袖領域の通信が支配的である．基本的に並列化には MPI を用いてプロセス並列で計算を行う．OpenMP と MPI を利用したハイブリッド並列については後半に一例をあげる．

4.1 FX10 の性能評価

FX10 では九大に導入された全ノード（768 ノード）を用いて性能評価を行った．コンパイラは Fujitsu Technical Computing Suite v1.0 を利用した．コンパイラオプションは以下の通りである．

-x3000 -Kfast, SPARC64IXfx, nomfunc, noalias=s, fsimple, prefetch_indirect, prefetch_strong, noparallel, array_private

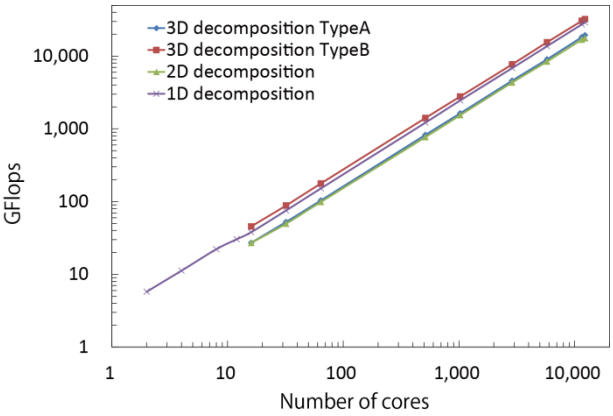

図 4　FX10 における MHD コードの実効性能
Figure 4　Performance of MHD code on FX10

図 4 に MHD コードの FX10 における実効性能を載せる．領域分割の種類により最低並列数が変わるため，利用コア数は異なっている．FX10 では 2 次元領域分割，3 次元領域分割の性能が悪く，ベクトル長の長い 1 次元領域分割手法の性能が高いことが分かる．さらにキャッシュヒットを考慮した 3 次元領域分割の Type B が 4 つの結果の中で最も性能が出ていることが分かり，キャッシュチューニングが効くことが分かる．後述するが，これは今までの POWER 系，SPARC 系の傾向と同じ結果である[5]．最大実行性能としては，全ノード利用時に 37TFlops，実行効率 20%となった．現状では配列の低次元化，変数の再利用率向上などのチューニングを施すと，22%程度の実行効率まで達成できている．また 4 ノードを利用して，OpenMP と MPI による Hybrid 並列を行った結果，Flat MPI 並列の実行効率から 2%程度低い結果となった．

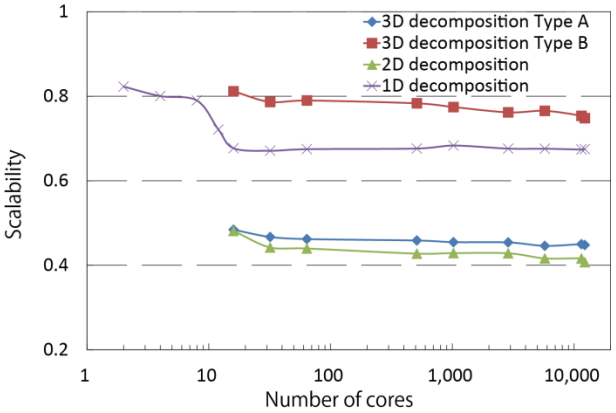

図 5　FX10 における MHD コードのスケーラビリティ
Figure 5　Scalability of MHD code on FX10

次に，FX10 の並列性能を見るために，図 5 にシリアル実行に対するスケーラビリティを載せる．1 次元領域分割において 10 コア利用付近で，大きな性能劣化が目立つが，これはノード内並列から，ノード間並列に変わる並列数であり，並列数が少ない場合，メモリバンド幅を共有するコアが少ないため，一般的に実効性能が出やすく，スケーラビリティの低下が見えやすい．16 並列を超えたスケーラビリティはどれもほぼ直線的な変化をしており，並列数の増加による並列性能の劣化はほぼ見えていない．FX10 はいわゆる Tofu インターコネクトを利用しており，ノード間通信性能が高い．

実際に並列化に伴う通信性能はどうなっているのかを確かめるため，袖通信にかかる時間，その通信のために配列データを格納する時間を計測すると，通信自体は実効性能に対して~1%の性能低下を与えている一方で，配列格納が 3%程度の性能低下をもたらしていることが分かった．これは後述する CX400 や XE6 では見えない現象であった．調査した結果，これは L2 キャッシュのスラッシングが原因と思われるが，解決方法はまだ見つかっていない．

4.2 CX400 の性能評価

本測定期間は CX400 の運用開始前の作業中であり、ノード障害があったため，九大に導入された全ノード（1476 ノード）は利用できず，1470 ノードを用いて性能評価を行った（正式稼働前）．コンパイラは Fujitsu Technical Computing Suite v1.0 を利用した．コンパイラオプションは以下の通りである．

-x3000 -Kfast, nomfunc, noalias=s, fsimple, noparallel

まず，図 6 に MHD コードの CX400 における実効性能を載せる．CX400 では各領域分割での性能差がほぼ見えず，キャッシュヒットを考慮した手法（Type B）だけが性能が出ていないことだけがはっきり見えている．これらから，

CX400 ではキャッシュチューニングはそれほど効果が無く，むしろベクトル的チューニングの方がより効果があることが分かる．これは今までの X86 系 CPU と同じ傾向である[5]．また 10,000 コアを超えた辺りで，実行性能に変化が見えているが，これは高並列時において，InfiniBand の動作が不安定のため，CX400 の挙動が安定しておらず，性能にぶれが出ることによる．最大性能は 1 次元領域分割時に 23,520 コアを利用し，100TFlops の実効性能，20%の実行効率だった．FX10 と同様にチューニングを施して，現状では 21%の実行効率を得ている．また 4 ノードを利用して，OpenMP と MPI による Hybrid 並列を行った結果，Flat MPI 並列の実行効率より，2%程度高い結果を得ており，スレッド並列の性能の高さが見える結果となっている．

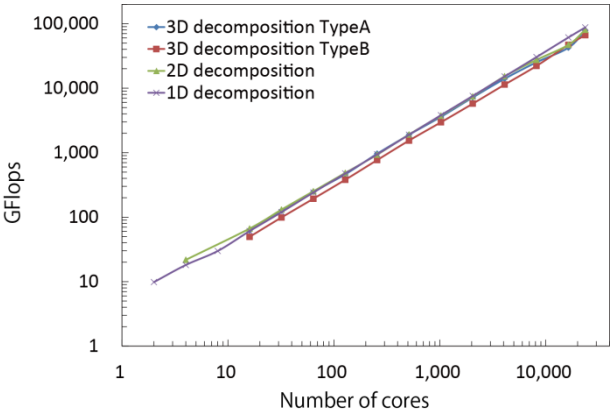

図 6　CX400 における MHD コードの実効性能
Figure 6　Performance of MHD code on CX400

次に図 7 にシリアル実行に対するスケーラビリティを載せる．低並列下では FX10 と同様の変化だが，並列数が上がるにつれて，スケーラビリティの低下と大きなぶれが見えてくる．CX400 は FX10 と違い InfiniBand を利用しているが，理論帯域幅は CX400 の方が高いにもかかわらず，実効並列性能は FX10 の方が高い．またスケーラビリティのぶれは前述の通り，InfiniBand の動作が不安定のために見えている．九州大学の CX400 ではノード群を 256 ずつのブロックに分け、ブロック内に対し、ブロック間のバンド幅が狭いため、リンクあたりのバンド幅を活かした通信が行えず､通信性能が低下することが考えられる.このため、ブロック間通信では帯域が狭くなり、衝突が発生しやすくなるが、実際には微妙な通信タイミングで衝突が発生したり、しなかったりすることが、性能ばらつきの原因の一つとして考えられる．

また FX10 で見られた通信に伴う配列格納による性能劣化を調べたところ，通信自体で 1.4%の性能低下，配列格納により 0.6%の低下が見られた．

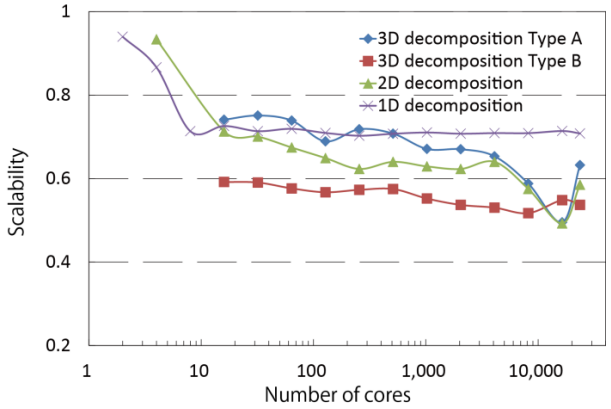

図 7　CX400 における MHD コードのスケーラビリティ
Figure 7　Scalability of MHD code on CX400

表 3　様々な計算機システムにおける性能の傾向
Table 3　Performance trend of various computer systems

	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
SX-8R	8/8	0.08	0.28	10.0	28	1D	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)
FX1	1024/256	2.08	10.24	8.1	21	3D_B	SPARC64VII
SR16000/L2	1344/672	5.38	25.27	8.0	21	3D_B	POWER6
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)
XE6	8192/512	14.16	81.92	27.7	17	1D or 2D	Opteron (Interlagos)
FX10	76800/4800	234.59	1135.41	48.9	21	3D_B	SPARC64 IXfx
CX400	23616/2952	104.23	510.11	35.3	20	3D_A	Xeon (Sandy Bridge)

4.3 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 3 にまとめた．ベクトル機はベクトル長が長い，1 次元，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，SPARC 系ではキャッシュヒットを考慮した 3 次元領域分割の Type B が最適になっている．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．

実行効率はベクトル機が高いという傾向だったが，最近のスカラチップは実行効率が上がってきており，Westmere の Xeon ではベクトルチップと同様の実行効率を達成しており，AVX により SIMD が倍になった Sandy Bridge Xeon を搭載する CX400 においても 20%程度の実行効率を達成しており，ベクトルチップの優位性が薄れてきているのが分かる．さらに 1CPU 当たりの実効性能をそれぞれ比較すると FX10 と CX400 では SX-9 を上回ることが分かり，特に FX10 では 1.4 倍の性能となっている．

5. まとめ

宇宙プラズマの研究に使われている MHD コードを利用して，九州大学に新しく導入された FX10 と CX400 の性能評価を行った．FX10 において，キャッシュヒットを考慮した 3 次元領域分割を用いて実効性能約 40TFlops（実行効率約 22%）を達成している．並列数が上がることによる並列性能の劣化は見られなかったが，通信に伴う配列の格納に時間がかかっており，性能劣化に繋がっていることが分かった．

CX400 では 1 次元領域分割を用いて，実効性能約 100TFlops（実行効率 21%）を達成している．InfiniBand の挙動が不安定のため，並列数増加に伴う並列性能の劣化，スケーラビリティのぶれが見えている．

今回の結果を様々な計算機システムと比較したところ，今回の結果は今までの傾向と同じで有り，さらにベクトルチップである SX-9 の 1CPU 性能を超えたことが分かった．

今回の結果により FX10, CX400 における最適化の傾向，性能を下げる要因が明らかになったため，今後はそれらを用いて最適化を加えてそれぞれ 25%に近い実効性能の達成を目指す．

謝辞 本研究の計算結果は九州大学情報基盤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.