Vol.2013-ARC-207 No.16
Vol.2013-HPC-142 No.16
2013/12/16

実アプリを用いたさまざまなアーキテクチャからなる計算機システムの性能評価

深沢圭一郎[†1, †2] 片桐孝洋[†3] 大宮学[†4] 江川隆輔[†5] 大島聡史[†3]
青木尊之[†6] 下川辺隆史[†6] 荻野正雄[†7] 岩下武史[†8] 東田学[†9]

現在いわゆるスーパーコンピュータと呼ばれる大規模計算機システムは x86 系，SPARC 系，POWER 系，ベクトル系，GPU 系などさまざまなアーキテクチャから構成されている．さらに近年では ARM 系や MIC などから構成される新しいスーパーコンピュータシステムも出てきている．これらのコンピュータシステムでは計算コアのアーキテクチャが異なるため，アプリケーションによってはそのシステムに対して向き不向きがあり，また性能チューニングも各アーキテクチャにより基本的には異なる．そのため，今までと異なるコンピュータシステムにアプリケーションの移植を行うことは非常に手間のかかる作業となっている．そこで本研究では，JHPCN-HPCI システムで利用可能である x86 系，SPARC 系，POWER 系，ベクトル系といった異なるアーキテクチャのコンピュータシステムを利用し，システムの実性能評価を行い，アーキテクチャ毎に性能がどのようになるか調べた．アプリケーションとしては実際に研究に用いられている電磁流体コードを用いて，いくつかの最適化を施したケースを評価した．その結果，x86 系ではベクトル系に効くベクトル化が効果的であり，SPARC 系と POWER 系ではキャッシュの効率的な利用が効果的となった．また，効果的な結果とそうで無いものを比べると倍程度の性能差があった．電磁流体コードは中性流体に電磁力を含めたコードであるため，本研究の結果は流体コードにも効果的であると考えられる．

Performance Evaluation of Computer Systems Consisted of Various Architectures with Scientific Application

KEIICHIRO FUKAZAWA[†1, †2] TAKAHIRO KATAGIRI[†3] MANABU OMIYA[†4]
RYUSUKE EGAWA[†5] SATOSHI OSHIMA[†3] TAKAYUKI AOKI[†6] TAKASHI SHIMOKAWABE[†6] MASAO OGINO[†7] TAKESHI IWASHITA[†8] MANABU HIGASHIDA[†9]

Recent supercomputer systems consist of various architecture such as X86, SPARC, POWER, Vector and GPU. In addition the computer systems which consist of MIC and ARM are appeared. Some applications are better suited for these computer systems or not due to the different architectures of calculation core they have. The way of performance optimization is also difference depending on the architecture. Thus it is hard to introduce applications running on a computer system to another computer system. In this study we evaluate the effective performance of various architectures computer systems using an MHD (magnetohydrodynamic) simulation code. The MHD simulation code used in this study solves the planetary magnetosphere in the space plasma. We evaluate two types of optimization, which are the vector suited and effective cache-hit. As the results, we found the vector suited optimization is effective to the X86 architecture computer systems and the cache hit optimization is suited for the SPARC and POWER architectures. There is the twice difference of performance between the results of effective optimization and not. The MHD simulation code is a kind of fluid code (including the magneto-electric force) so that it is general that these results are performed in the fluid codes.

1. はじめに

現在いわゆるスーパーコンピュータと呼ばれる大規模計算機システムは x86 系，SPARC 系，POWER 系，ベクトル系，GPU 系などさまざまなアーキテクチャから構成されている．さらに近年では ARM 系や MIC などから構成される新しいスーパーコンピュータシステムも出てきている．これらのコンピュータシステムでは計算コアのアーキテクチャが異なるため，アプリケーションによってはそのシステムに対して向き不向きがあり，また性能チューニングも各アーキテクチャにより基本的には異なる．

2000 年以前では大規模計算機システムとしてはベクトル型 CPU が主流であったが，近年ではスカラ型 CPU によるクラスタ型超並列計算機が主流となっている．しかし，

†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 東京大学情報基盤センター
Information Technology Center, The University of Tokyo
†4 北海道大学情報基盤センター
Information Initiative Center, Hokkaido University
†5 東北大学サイバーサイエンスセンター
Cyberscience Center, Tohoku University
†6 東京工業大学学術国際情報センター
Global Scientific Information and Computing Center, Tokyo Institute of Technology
†7 名古屋大学情報基盤センター
Information Technology Center, Nagoya University
†8 京都大学学術情報メディアセンター
Academic Center for Computing and Media Studies, Kyoto University
†9 大阪大学サイバーメディアセンター
Cybermedia Center, Osaka University

Vol.2013-ARC-207 No.16
Vol.2013-HPC-142 No.16
2013/12/16

これまでのベクトル型 CPU による並列計算機とは異なり，スカラ型 CPU で高い実効性能を達成することは容易ではなく，また並列数が格段に上がったために高い並列効率を達成することが容易ではないという 2 つの問題が生じている．例えば，これまで地球シミュレータなどのベクトル型 CPU において高い実効効率を誇っていたコードが，スカラ型 CPU においては実効性能が低い例は少なくない．また，スカラ型 CPU であっても，CPU アーキテクチャが異なれば，チューニングの手法も異なるため，同じ最適化で実行効率が常に高くなるとは限らない．一方で，近年開発されたアプリケーションは x86 型の計算機システムに最適化されていることが多く, ベクトル型 CPU では高メモリバンド幅を使い切れないという問題もある．

並列化の問題に関しては，1,000 個以上の CPU コアを用いた計算を日常的に実行できる環境が日本にあまり存在しないために，どのシステムにおいても並列化のスケーラビィティが保障されるコードの開発が困難となっている．

そのため, 今までと異なるコンピュータシステムにアプリケーションの移植を行うことは非常に手間のかかる作業となっている．また，科学アプリケーションを利用して研究を行っている計算機ユーザにとって移植作業は時間がかかるだけで，その作業自体は成果にならないため，移植が行われにくい状況にある．

そこで本研究では，x86 系，SPARC 系，POWER 系，ベクトル系といった異なるアーキテクチャのコンピュータシステムを利用し，システムの実性能評価を行い，アーキテクチャ毎に性能がどのようになるか調べる．アプリケーションとしては実際に研究に用いられている電磁流体（MHD）コードを用いて，いくつかの最適化を施したケースを評価する．

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 コードを使用した様々なアーキテクチャからなる計算機システム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．より詳細な磁気圏の紹介については，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．

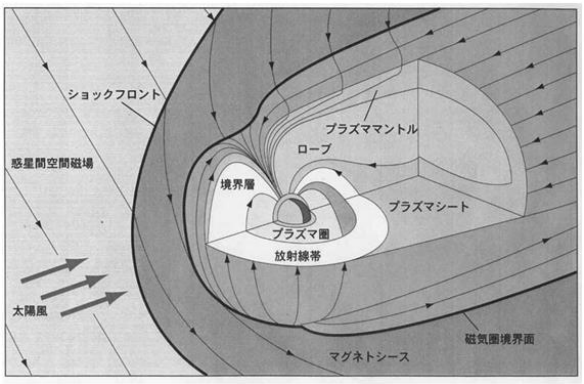

図 1　地球磁気圏構造

Figure 1　Configuration of Terrestrial magnetosphere

宇宙プラズマの密度は非常に低いために，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，宇宙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．しかしながら，Vlasov 方程式はこのように多くの成分からなる 6 次元非線形方程式であり，現在の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 MHD 方程式が，惑星磁気圏などの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．MHD 方程式は(1)式，Vlasov 方程式のそれぞれ 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

Vol.2013-ARC-207 No.16
Vol.2013-HPC-142 No.16
2013/12/16

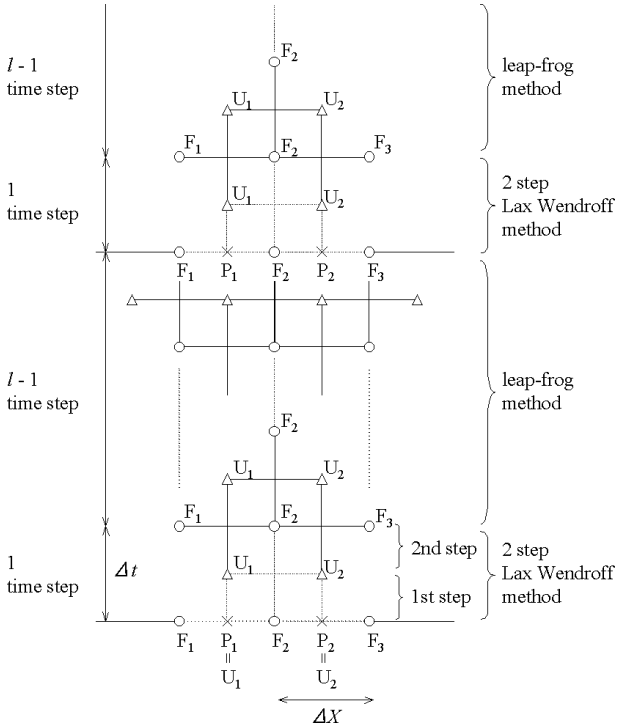

図 2　Modified Leap Frog 法の計算スキーム

Figure 2　Scheme of Modified Leap Frog method

とで得られ，以下のようになる．

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

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. シミュレーションモデル

3.1 数値計算法

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. 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もある[5].

3.2 評価モデル

並列化には MPI を使用する（Flat MPI）．並列化手法としては MHD 方程式で解く 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るが，本性能評価では 3 次元領域分割を用いて評価を行う（1 システムだけ例外あり）.

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 $f(i, j, k, m)$ （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 $f(m, i, j, k)$ （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[5]．そのため，本性能評価においてもこの配

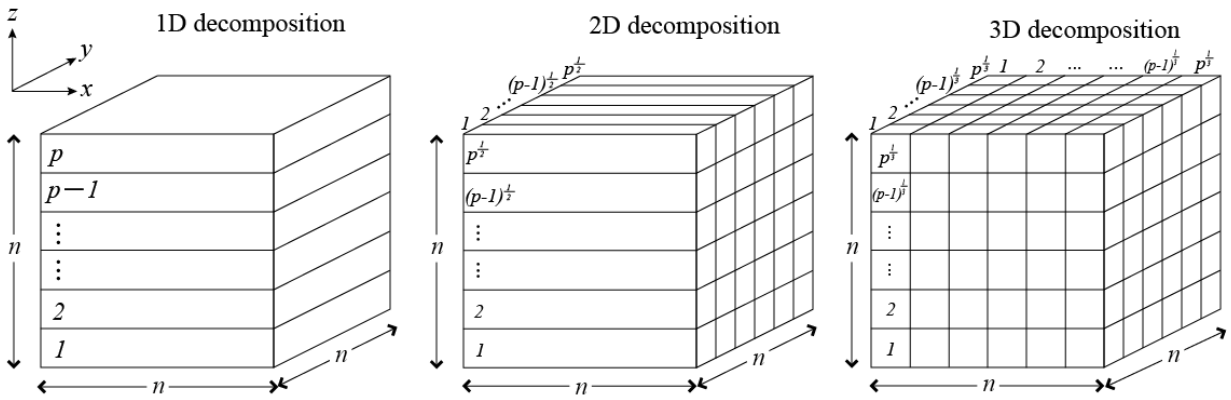

図 3　1 次元，2 次元，3 次元領域分割の模式図

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

表 1　計算機システムの構成

Table 1　Character of computer systems

System	SR16000/M1	SX-9	FX10	XE6	CX400
CPU	POWER7 8 cores 3.83 GHz 245.1 GFlops L2: 256 KB/core L3: 32 MB/CPU	Vector 1 core 3.2 GHz 102.4 GFlops ADB: 256 KB	SPARC64 IXfx 16 cores 1.848 GHz 236.5 GFlops L2: 12 MB/CPU	Opteron62xx 16 cores 2.5GHz 160 GFlops L2: 2 MB/2cores L3: 16 MB/CPU	Xeon E5-26xx 8 cores 2.7 GHz 172.8 GFlops L2: 256 KB/core L3: 20 MB/CPU
Number of CPU　per node	4	16	1	2	2
Memory size per node	128 GB	1 TB	32 GB	64 GB	128 GB
Memory bandwidth per node	512 GB /s	4096 GB/s	85GB/s	102.4 GB/s	51.2 GB/s
Number of nodes	176	18	4800	940	1476
Inter-node connection	20 GB/s × 2	128 GB/s × 2	Tofu Interconnect (5GB/s)	Gemini, 9.3GB/sec or 4.6GB/sec	InfiniBand FDR (6.78GB/s)
Theoretical performance	172 TFlops	29.5 TFlops	1.1 PFlops	300.8 TFlops	510.1 TFlops

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

本性能評価では，北海道大学 SR16000/M1（POWER 系），東北大学 SX-9（ベクトル系），東京大学 FX10（SPARC 系），京都大学 XE6（x86 系 Opteron），九州大学 CX400（x86 系 Xeon）を利用した．それぞれの計算機システムの構成は表 1 の通りである．

今回の性能評価では 64 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 192 MB/コアを追加で使用した．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．

4.1 日立 SR16000/M1 の性能評価

北海道大学 SR16000/M1 では，最大 4,096 コア（128 ノード）を用いて性能評価を行った．コンパイラは日立最適化 FORTRAN を利用した．コンパイラオプションは以下の通りである．

-model=M1 -Oss -parallel=0 -divopt -pvfunc=0 -looptiling -nolimit -noscope -loglist

図 4 に MHD コードの SR16000/M1 における実効性能を載せる．横軸が利用コア数，縦軸が実効性能領を示す．POWER7 では SMT（Simultaneous Multithreading）が利用できるため，SMT を利用して，1 コアに 2 プロセス立ち上げた場合も評価した．図 4 より SR16000/M1 ではキャッシュヒット効率を高めた Type B の配列が Type A よりも性能が良い（SMT の有無に関係なく）．このとき，SMT 無しの場合で 14.2 TFlops（実行効率 11.4 %），SMT 有りで，19.5 TFlops（15.5 %）を達成した．Type A は SMT 無しで，6.4 TFlops と Type B に比べて，半分未満の性能にとどまっている．

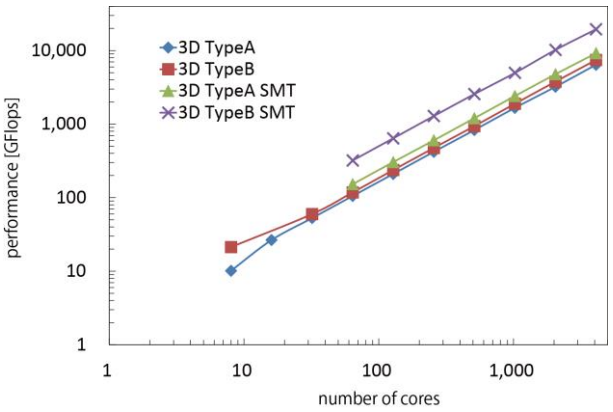

図 4　SR16000/M1 における MHD コードの実効性能

Figure 4　Performance of MHD code on SR16000/M1

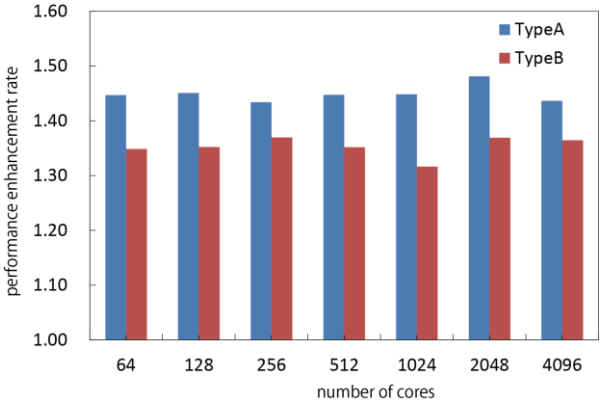

図 5　SMT 利用時の性能向上率

Figure 5　Performance enhancement rate by SMT

また，SMT により性能がどの程度向上するか確認するため，図 5 に SMT 利用時の性能向上率をまとめた．利用コア数によって性能向上率にばらつきがあるのは予想外だが，Type A で約 1.45 倍の向上，Type B で 1.35 倍の向上となっ

ている．1,024 コア時だけ向上率が低い．SMT は物理コア上に論理コアを作るイメージなので，実行効率が悪い（性能を使い切れていない）Type A の場合により性能向上が見られると考えられる．

詳細説明は省くが，通信のまとめ，非同期通信，OpenMP の導入で，3%の実行効率向上が可能であった．

4.2 NEC SX-9 の性能評価

東北大学 SX-9 では，最大 64 コア（4 ノード）を用いて性能評価を行った．コンパイラは NEC FORTRAN90/SX を利用した．コンパイラオプションは以下の通りである．

-Chopt -size_t64 -Wl,-h, 8T_memlayout

図 6 に SX-9 における MHD コードの実効性能を載せる．横軸が利用コア数，縦軸が実効性能領を示す．本研究では 3 次元領域分割 Type A と B の評価だけだが，1 次元，2 次元領域分割の性能が良いので，参考までに結果を図 6 に載せる．図 6 より SX-9 では 3 次元領域分割 Type A の性能（935.2 GFlops）が Type B（449.7 GFlops）よりも明らかに高い．これは SR16000/M1 と逆の結果となっている．ただし，今回の評価ではコア当たりの利用配列が小さいため，3 次元領域分割ではループ長が長くなりにくいため，1 次元，2 次元領域分割に比べて性能が悪くなった．それぞれ最高で 1.9 TFlops（28 %）の性能を達成している．これも 3 次元領域分割の結果と倍程度差があり，ベクトル機の特徴に合わせた最適化が重要とわかる．

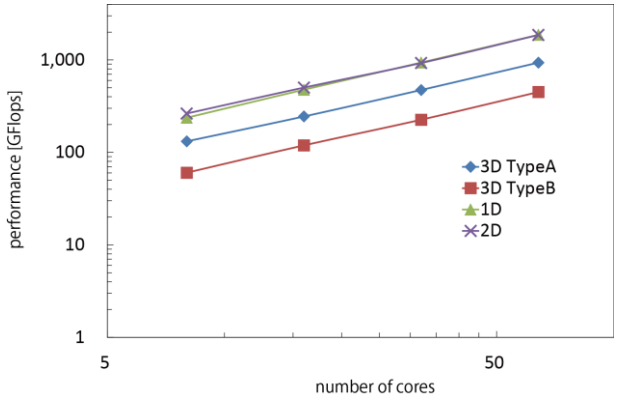

図 6　SX-9 における MHD コードの実効性能

Figure 6　Performance of MHD code on SX-9

4.3 富士通 FX10 の性能評価

東京大学 FX10 では，最大 76,800 コア（4,800 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 7 に MHD コードの FX10 における実効性能を載せる．横軸が利用コア数，縦軸が実効性能領を示す．FX10 では，SR16000/M1 と同様に Type B の方が Type A よりも性能が高い．最大で 76,800 コア利用時に 198.2 TFlops（17.5 %）を達成している．一方で，Type B では半分近くの性能である 125.9 TFlops となっている．ここでは詳細は述べないが，OpenMP と非同期通信を利用することで，5 %ほどの実行効率向上が可能であった[5].

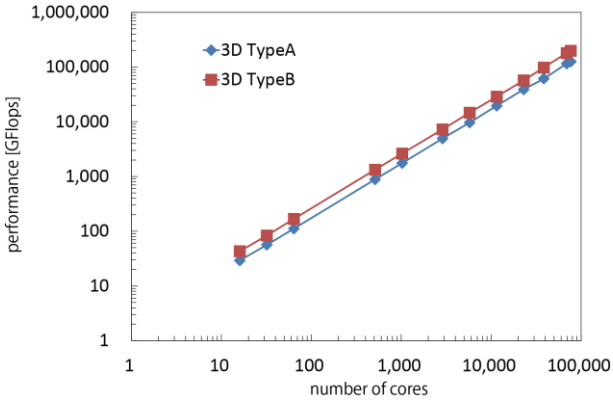

図 7　FX10 における MHD コードの実効性能

Figure 4　Performance of MHD code on FX10

4.4 CRAY XE6 の性能評価

京都大学 XE6 では，最大 8,192 コア（256 ノード）を用いて性能評価を行った．コンパイラは CRAY コンパイラを利用した．コンパイラオプションは以下の通りである．

-O3,ipa5,aggress -em -h pic -dynamic -h noomp -h msgs -h negmsg

図 8 に MHD コードの XE6 における実効性能を載せる．横軸が利用コア数，縦軸が実効性能領を示す．XE6 では今までの結果より Type A と Type B に性能の差がないが，Type A の性能（12.3 TFlops）が Type B（9.9 TFlops）より良かった．さらに詳細は示さないが，2 次元領域分割で性能が最も良くなり，最大で 14.3 TFlops を達成している．このことから XE6，Opteron 6000 シリーズはベクトル機向け最適化が効果的と言える．

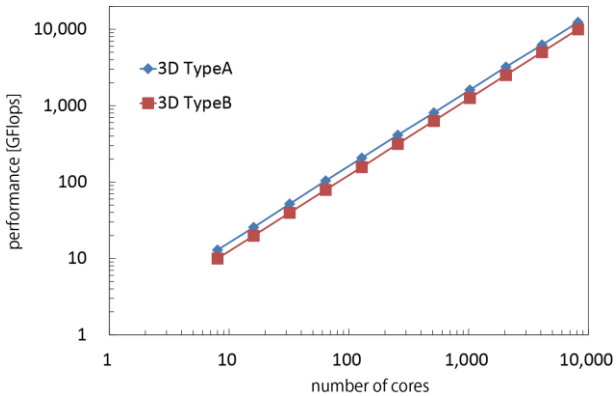

図 8　XE6 における MHD コードの実効性能

Figure 8　Performance of MHD code on XE6

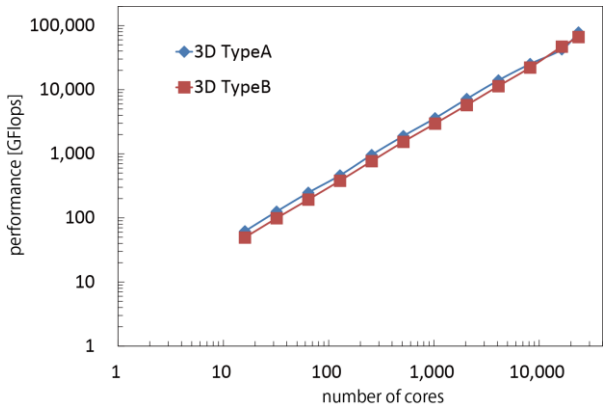

図 9　CX400 における MHD コードの実効性能
Figure 9　Performance of MHD code on CX400

4.5 富士通 CX400 の性能評価

九州大学 CX400 では，最大 23,616 コア（1,476 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, nomfunc, noalias=s, fsimple, noparallel

図 9 に MHD コードの CX400 における実効性能を載せる．横軸がコア数，縦軸が実効性能領を示す．この計算機システムにおいても Type A と Type B の性能差が少ない．図ではほぼかぶって見えるが，Type A で最高性能を達成し，78,0 TFlops となった．Type B では 66.3 TFlops になっている．この計算機システムは 256 ノードを一グループにし，グループ内と外で通信帯域に差があるため，利用コア数が延びるほど，スケーラビリティが下がっている様子が見える．

4.6 他システムとの比較

今回行った 5 つの計算機システムの性能評価結果と今までに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[5，6，7]．この表では FX10 は Hybrid MPI＋非同期通信の結果，XE6 は 2 次元領域分割の結果，CX400 は非同期通信の結果から値を持ってきている．

表を見ると，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．さらに近年発表された x86 系アーキテクチャを持つ MIC である Xeon Phi でもベクトル的な最適化が効果的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きており，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

表 2　様々な計算機システムにおける性能の傾向[5，6，7]
Table 2　Performance trend of various computer systems [5，6，7]

	Core/CPU	Rpeak [TFlops]	Rmax [TFlops]	Rpeak /CPU [Gflops]	Efficiency [%]	Suitable domain decomposition	CPU architecture
SX-8R	8/8	0.08	0.28	10.0	28	1D	Vector
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
SR16000/M1	4096/512	19.49	125.50	38.1	16	3D_B	POWER7
Xeon Phi 5110P	60/1	0.049	1.01	49.0	5	3D_A	Knights Corner

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.

CPU 当たりの性能を比べてみると，今回性能を評価した FX10 と Xeon Phi ではほぼ同じ性能を持っていることが分かる．この Xeon Phi は Native で利用している．詳細は参考文献 7 を参照されたい．また CX400 は K より CPU 当たりの性能は高く，SX-9 の 1CPU よりも高い性能となっている．

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，様々なアーキテクチャを持つ計算機システムの性能評価を行った．評価では一般的に利用されるベクトル向けの配列並びとキャッシュヒットが効率的に行われるような配列並びを用いた．その結果，SR16000/M1 や FX10 ではキャッシュヒットの最適化が有効であり，x86 系である XE6 や CX400 ではベクトル向け最適化が効果的だとわかった．ベクトル機ではそれらの配列並びから更にベクトル長を長くするような最適化を行わないと，性能が出ないことも確かめられた．特に SR16000/M1，FX10 や SX-9 では配列の並びが異なるだけで性能が 2 倍近く違うことが示され，最適化の重要性が確認できる．

今までに性能測定を行った別のシステムと比べたところ，上記の傾向は同じアーキテクチャであれば共通のように見える．ただし実行効率は大きく変動している．

本研究では詳細な最適化を各システムに行っていないが，今後詳細な最適化を行い，より各アーキテクチャの最適化情報を蓄積し，公開していく予定である．

謝辞 本研究の計算結果は学際大規模情報基盤共同利用・共同研究拠点（JHPCN），東京大学情報基盤センター大規模 HPC チャレンジ，九州大学情報基盤研究開発センター先端的計算科学研究プロジェクトにより計算機システム利用の支援を受けた．

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., 38, 9, 2254, 2010.

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) 深沢，岡，”電磁流体コードを用いた Xeon Phi の性能評価”, 第 141 回 HPC 研究会，2013-HPC-141 No.5, 2013.