

# 惑星磁気圏 MHD シミュレーションコードによる ThunderX2 ARM プロセッサの性能評価

深沢 圭一郎<sup>†</sup>

**概要:** ThunderX2 はサーバ向け ARM プロセッサであり、近年 HPC サーバにもいくつか採用されている。ARM プロセッサは HPC 向け SIMD 拡張版プロセッサが次世代スーパーコンピュータ富岳に採用されているように、今後 HPC 分野での採用が広がる可能性がある。しかしながら、ThunderX2 では 128bit 幅の SIMD 演算までしか行えないため、現在の Xeon や Xeon Phi に比べて理論性能が劣り、基本的ベンチマークでは、高い性能を発揮していない。そこで本研究では、実際に研究に利用されている惑星磁気圏 MHD シミュレーションコードを用いて、ThunderX2 の性能測定を行い、他の CPU の性能と比較することで ThunderX2 の実効性能を評価する。

**キーワード:** ARM プロセッサ, 性能評価, MHD シミュレーション

## Performance Evaluation of ThunderX2 ARM Processor with MHD Simulation of a Planetary Magnetosphere

KEIICHIRO FUKAZAWA<sup>†</sup>

**Abstract:** ThunderX2 is Armv8 based processor designed for the computer server and has been introduced to several supercomputer systems recently. As Fugaku next-generation supercomputer system in Japan adopts the ARM processor with extended SIMD for HPC, the HPC computer system consists of the ARM processor will broaden. However, ThunderX2 can execute the only 128bit SIMD operations compared to the 512bit SIMD operations by the latest Xeon and Xeon Phi. Thus, the ideal performance of ThunderX2 is not high and the basic benchmark on ThunderX2 does not achieve high performance. In this study using the magnetohydrodynamic simulation code for planetary magnetosphere (not benchmark code), the real performance of ThunderX2 is evaluated and compared to the other CPU real performance.

**Keywords:** ARM processor, Performance evaluation, MHD simulation

### 1. はじめに

ベクトル型計算機から始まったスーパーコンピュータは近年では主にスカラ型計算機、特に x86 アーキテクチャにより構成されている。x86 アーキテクチャを利用した Xeon 系 CPU はプロセス微細化技術やリーク電流の問題もあり、周波数の向上が難しくなった結果、コア数の増加や同時演算数の増加により、CPU 性能の向上を達成している。例えば、Xeon Phi KNL では、1.5GHz 程度の周波数に 60 を超える CPU コア、倍精度での同時演算数が 32 であり、その結果、理論性能は 3 TFlops に達する。Xeon であっても Skylake (SKL) 世代であれば、2GHz 程度の周波数に同時演算数が 32、コア数が約 16 となっており、2 ソケットでの理論性能は Xeon Phi KNL と同程度になっている。

一方、モバイル系で主に利用されている ARM プロセッサは、その主な用途のため、計算性能を制限することで消費電力を下げており、HPC のような高性能計算には利用されてこなかった。このような中 Cavium (現 Marvell) は、計算性能を高めたサーバ向け ARM プロセッサ ThunderX2

をリリースしている[1]。ThunderX2 は、近年 HPC サーバにもいくつか採用されており、また、HPC 向け SIMD 拡張版 ARM プロセッサが次世代スーパーコンピュータ富岳に採用されていることから、今後 HPC 分野での採用が広がる可能性がある。しかしながら、ThunderX2 では 128bit 幅の SIMD 演算までしか行えないため、現在の Xeon や Xeon Phi に比べて理論性能が劣り、基本的ベンチマークでは、それほど高い性能を発揮していない[2]。計算機システムの理論性能はカタログスペックにより分かるが、実際にアプリケーションを動かした場合にどのような性能になるのかは、予測が難しい。

そこで、本研究ではこれまでに様々なスーパーコンピュータで性能評価を行ってきた電磁流体 (MHD) シミュレーションコード[3]を用いて、ThunderX2 の性能評価を行う。MHD シミュレーションは通常の流体シミュレーションに電磁場の効果を考慮したシミュレーションになっており、本性能評価の結果は流体系のアプリケーションに広く応用でき、また、これまでに評価してきた計算機システムの性能と比較することで、現実的な計算性能を見積もることも

<sup>†</sup> 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

可能と考えられる。

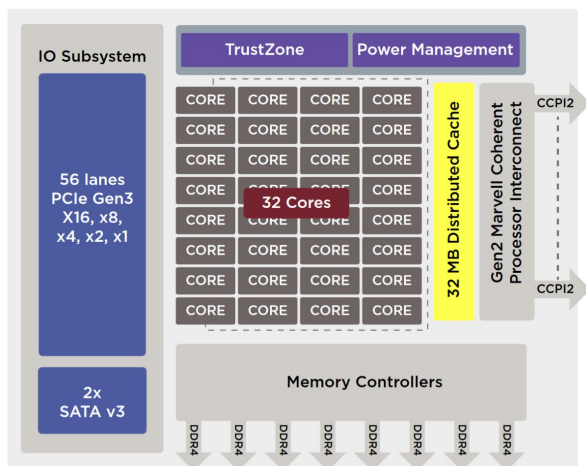


図1 ThunderX2 の構造[1]

Figure 1 Architecture of ThunderX2[1]

本研究報告の構成は以下の通りである。第2章では、ThunderX2 搭載計算機システムについて説明し、第3章では MHD シミュレーションコードについて説明をする。第4章で性能評価の結果を述べ、その結果と他システムとの比較を第5章で行い、最後に研究のまとめをする。

## 2. ThunderX2 計算機システム

本性能評価で利用する ThunderX2 搭載計算機システムは、次世代のスーパーコンピュータの評価を目的として、2019年に試験的に京都大学学術情報メディアセンターに導入されたシステムであり、4ノードの HPE Apollo 70 System により構成される。この計算機システムは図1に示されるような ARM プロセッサである Cavium ThunderX2 を搭載し、1ノード当たり 2CPU, 64 コア(理論演算性能 1,124GFlops), 128GB メモリ (バンド幅 341.2GB/s) からなる。システムの諸元を表1に示す。

ThunderX2 は、同時演算数が 32 である SKL Xeon (Gold 以上) や Xeon Phi KNL とは異なり同時演算数は 8 となっており、理論演算性能が相対的に低くなっている。また、CPU 当たりのメモリチャンネルは 8 となっており、メモリバンド幅は Xeon よりも高い。これらにより、理論演算性能とメモリバンド幅の比である B/F 値が 0.30 と Xeon 搭載計算機システムに比べて高くなっている。

HPC 系のベンチマーク結果は、ThunderX2 の White paper[2]で紹介されており、SKL Xeon との比較がなされている。表2にその結果を示すが、ThunderX2 は SKL Xeon と比較して、メモリチャンネルの数増加によるバンド幅向上効果、SIMD 幅が 128bit と低いことによる実行効率の上昇効果が見える結果となっている。このベンチマークで利用された計算機システムの詳細は不明だが、SKL Xeon Gold

表1 ThunderX2 搭載計算機システムの諸元

System	HPE Apollo 70 System	
Node	CPU	Cavium ThunderX2 CN9980-2200 × 2 /node
	Num. of core	32 cores /CPU
	Frequency	2.2 GHz
	Cache	L1 : 32 KB, L2 : 256 KB/core L3 : 32 MB/CPU
	Rpeak	1,124 GFlops /node (倍精度)
	Memory	DDR4 128 GB /node
	Bandwidth	341.2 GB/s /node
	B/F	0.30
Num. of node	4 nodes	
	Rpeak	4.5 TFlops
	Node comm.	InfiniBand EDR 100 Gbps

表2 ThunderX2 のベンチマーク結果 [2]

Workload	ThunderX2	SKL Xeon Gold 6148
STREAM	251 GB/s	198 GB/s
HPCG	35 GFlops	36 GFlops
HPL	92 %	72 %

(ベース周波数 2.4GHz, 20 コア) 2 ソケットサーバと ThunderX2 (32 コア) 2 ソケットサーバの比較となっている。このように ThunderX2 のコア数が SKL Xeon に比べて 1.6 倍あることで、計算性能の差があまり見えないと考えられる。

## 3. MHD シミュレーションコード

宇宙空間は真空とされているが、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。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。この太陽風が惑星の固有磁場と相互作用することで、惑星磁場が変形し、磁気圏と呼ばれる領域が形成される。磁気圏では、磁気嵐など様々な現象が起きており、その領域だけでなく、地上にも様々な形で影響を及ぼしている(図2参照)。このような現象は宇宙天気と呼ばれ、古くから研究が行われている。また、HPCI 萌芽課題の一分野として富岳を利用する数値シミュレーション研究も進められている。

このような宇宙プラズマ現象である宇宙天気を理解す

るためには、宇宙プラズマの振る舞いを記述する方程式、Vlasov-Maxwell 方程式を解く必要がある。これは、無衝突 Boltzmann 方程式と Maxwell 方程式から成る。Vlasov (無衝突 Boltzmann) 方程式は以下の形をとる。

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

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,  $f_s(\vec{r}, \vec{v}, t)$  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。

Vlasov 方程式はプラズマの振る舞いを正確に記述しているが、多くの成分からなる非線形方程式であり、近年の計算機システムを用いても、磁気圏全体をグローバルに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。MHD 方程式は以下のようになる。

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

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)、最後に磁場の誘導方程式となる。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[5]。

MHD 方程式を解く数値計算法としては、Modified Leap Frog (MLF) 法[3, 6]という数値計算法を使用する。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、本手法で採用する 2 次精度の中心空間差分では、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。MHD コードはこのような数値計算を Fortran で実装している。

並列化にはプロセス並列に MPI 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。領域分割には、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、MPI 通信に伴うデータのバック/アンパックや通信量自体が変わるため、それぞれの性能評価を行う。

一般的に B/F 値の低いスカラ CPU で性能を出すため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メモリアクセス時に、その周辺数 KB のデータをキャ

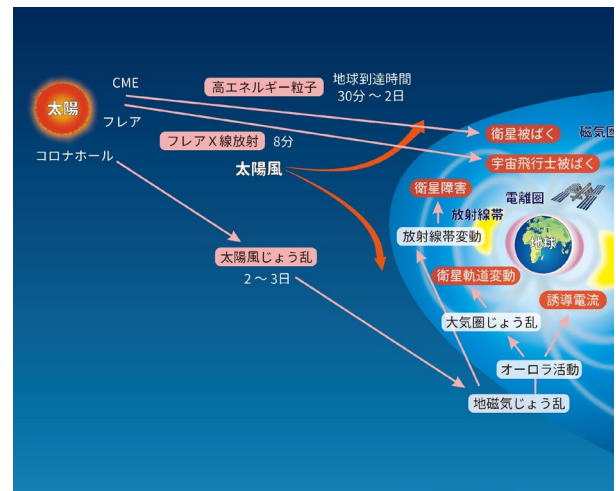


図2 宇宙天気現象[4]

Figure 2 Phenomena of Space Weather [4]

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ため、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  $f(x, y, z, m)$  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 Fortran では、 $f(m, x, y, z)$  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]。しかしながら、近年の Xeon 系 CPU のように SIMD 幅向上に伴いベクトル化が性能向上にとって重要な機構であるため、更に配列を  $f(x, m, y, z)$  と  $f(x, y, m, z)$  と定義した場合の性能評価も行う。

#### 4. 計算性能評価

ThunderX2 計算機システムでは、コンパイラとして Arm HPC Compiler 19.1.0 を利用し、MPI は HPE MPI 2.19 を利用し、性能評価を行った。計算サイズは基本的にはプロセス当たり、64MB (100<sup>3</sup>グリッド) となり、Weak scaling の評価である。計測は 5 回行い、その平均値を取った。評価用 MPI-OpenMP ハイブリッドコードが実行できなかったため、今回は Flat MPI 並列実行で評価を行った。

Arm HPC Compiler 19.1.0 では、Fortran を利用し、コンパイル (オプション含む) には下記を利用した。

```
mpif90 -f90=armflang -Ofast -mcpu=native
```

オプションの詳細は Arm Allinea Studio [8] のマニュアルを参照されたい。

図 3 に ThunderX2 計算機システムを利用して、3 種類 (1 次元, 2 次元, 3 次元) の領域分割を行った MHD コードの評価結果を示す。前述のように、プロセス当たりの計算量がどの領域分割においても  $(x, y, z) = (100, 100, 100)$  となるよ

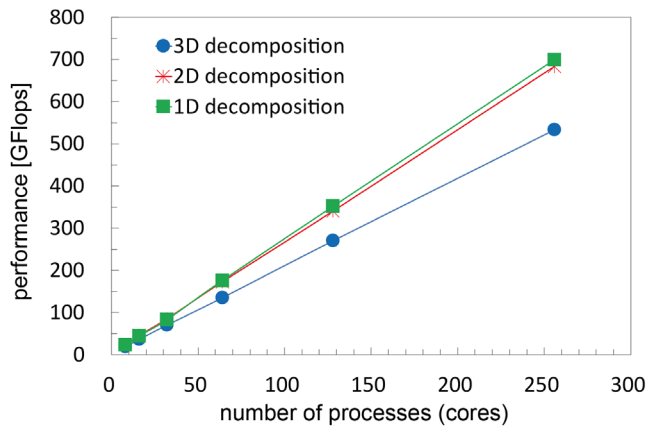


図3 3種類の領域分割によるMHDコードの性能  
Figure 3 Performance of MHD code with three domain decompositions

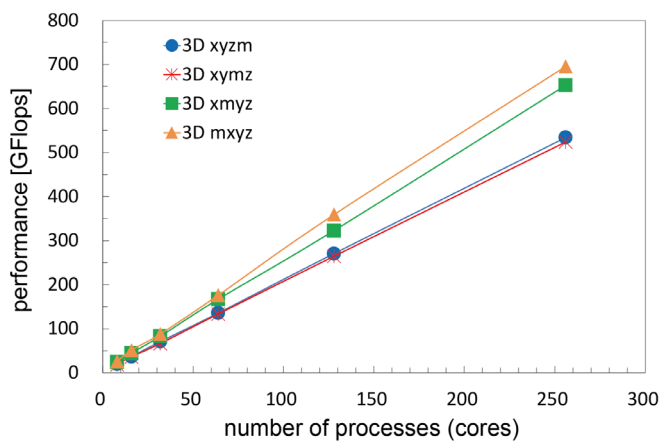


図4 配列並びが異なる場合のMHDコードの性能  
Figure 4 Performance of MHD code with different array order

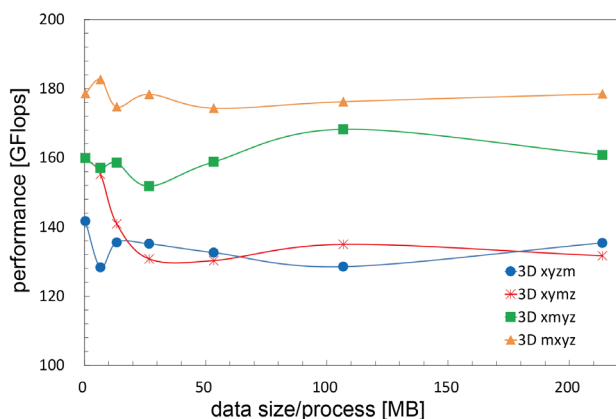


図5 異なる配列データサイズにおけるMHDコードの性能  
Figure 5 Performance of MHD code with different array data size

表3 各領域分割における袖通信時間 (パック/アンパック含む)

Table 3 Halo communication time including pack/unpack operation on each domain decomposition

	通信時間/step [sec]	通信時間/64step [sec]
1D	0.001901874	0.486879706
2D	0.003084081	0.789524734
3D	0.018849426	4.825453103

うに設定し、ここでは主に領域分割による性能の違いを見ている。1次元領域分割と2次元領域分割の結果はそこまで差が無いが、並列数が上がると性能差が見えてきており、256プロセスを利用した場合、1次元領域分割で700 GFlops (実行効率15.5%)、2次元領域分割で684 GFlops (同15.2%)となっている。一方で3次元領域分割では、CPUコアをすべて使わずコア当たりのメモリバンド幅が高い状態の8プロセスにおいても、1、2次元領域分割に比べ2.7%も実行効率が低く(14.0%)、並列数が上がるにつれて、計算性能にも大きな差が現れている。256プロセス利用で、534 GFlopsであり、約150 GFlopsの性能差が見て取れる。

今回の評価では、各プロセスでの計算に利用される配列の形状は同じであり、前述の性能の差は通信部分から来ていると考えられる。一般に、領域分割の次元が上がるほど通信量が減り、通信性能が向上すると考えられるが、次元数が増えることで、通信すべきデータのバック/アンパックが発生する。各領域分割のバック/アンパックを含む袖通信部分の時間を計測した結果を表3に示す。3次元領域分割だけが5~10倍程度時間がかかっており、この部分が性能に影響していると考えられる。

次に、配列の並びによるMHDコードの性能変化を評価した。図4に3次元領域分割において、前述のように配列の並びが異なる4つの場合にMHDコードがどのような性能を示したかを載せている。図中の3D xyzmは、図3の3次元領域分割の結果と同じであり、xyzmは配列の並びを表している(これは $f(x, y, z, m)$ を意味する)。大まかに見ると、性能が良いものと悪いものに二つに分かれた。3D mxyzとxmyzの性能が高く、3D xyzmとxymzの性能が低くなった。256プロセス利用で、最大性能は695 GFlops (実行効率15.4%)であり、最小性能は524 GFlops (同11.6%)と大きな差が現れた。性能が高い場合は、物理変数を示すmが最内に近く、いわゆるAoS (Array of Structure)形式であり、これまでの研究結果によりキャッシュヒット率が高い配列構造である[7]。

一方で低い性能を示した配列は、コード内の主な計算ループがxyzの三重ループであり、Fortran配列は最内からメモリアクセスをすることから、ベクトル長が長い構造をしていると考えられる。ThunderX2は近年のXeon系CPUや

表 4 様々な計算機システムにおける性能の傾向[7, 9, 10]

Table 4 Performance trend of various computer systems [7, 9, 10]

	Core/CPU	Rmax [TFlops]	Rpeak [TFlops]	Rmax /CPU [GFlops]	Efficiency [%]	Suitable domain decomposition	CPU architecture
<b>SX-ACE</b>	1024/256	29.20	65.50	114.0	45	3D xyzm	Vector
<b>K</b>	262144/32768	914.12	4194.30	27.9	22	3D mxyz	SPARC64 VIIIfx
<b>FX100</b>	16384/512	91.49	576.72	178.7	17	3D xyzm	SPARC64 XIfx
<b>CX400</b>	23616/2952	104.23	510.11	35.3	20	3D xyzm	Xeon (SandyBridge)
<b>HA8000</b>	23160/1930	83.42	500.26	43.2	17	2D xyzm	Xeon (IvyBridge)
<b>XC30</b>	448/32	1.37	16.49	42.8	8	2D xyzm	Xeon (Haswell)
<b>XC40</b>	1088/16	4.32	48.86	273.3	9	3D xyzm	Xeon Phi KNL
<b>Xeon Phi 5120</b>	60/1	0.08	1.00	84.0	8	3D xyzm	Xeon Phi KNC
<b>Tesla K20X</b>	896/1	0.15	1.31	153.3	12	3D xyzm	Kepler
<b>ITO-A</b>	72000/4000	470.10	6912.00	117.5	7	1D xyzm	Xeon (Skylake)
<b>ThunderX2</b>	256/8	0.70	4.50	86.9	16	3D mxyz	Arm v8

富岳で利用される ARM 系 CPU と比べ, SIMD 計算の bit 幅が少なく, この配列構造では性能が低くなる可能性がある。

最後に, 計算に利用する配列のサイズによる ThunderX2 の性能を評価した。これまでの結果からキャッシュヒット率が高い場合の性能が高いことが示唆されている。そこで配列のサイズを計算で利用するすべての配列がオンキャッシュになるようなサイズまで小さくしていくことで, 計算性能がどのように変化するかを調べた。図 5 では, 64 プロセス (1node) 利用時にプロセス当たりの総配列利用サイズを 0.4~213 MB (28~13,632 MB/node) と変化させた場合の配列構造が異なる MHD コードの性能を示している。図中の 3D xyzm は図 4 の場合と同じである。ThunderX2 の L3 キャッシュサイズは 32 MB/CPU であり, 最小の配列サイズでは L3 のサイズを下回っている。

計算サイズによって大きく性能が変化した配列構造は 3D xymz であり, 最もサイズが小さい (L3 にオンキャッシュサイズ) では, 3D xmyz と同じ性能となっている。3D xymz は図 4 の異なる配列構造の性能では最も性能が低い構造であったが, 同じく 2 番目に性能が高かった 3D xmyz の性能と同等になっている。最も配列サイズが小さい場合, z 方向には 10 要素あり, それを除いた場合の配列サイズを計算すると 43 KB となり (xmy と xym 分のサイズ), L2 キャッシュ (256 KB/core) に載るため, この二つの配列構造は結果的に同じ性能を示していると考えられる。

一方で, 3D xyzm の構造はサイズが小さい場合でもそれほど性能に変化がなく, 最小の場合に少し性能向上が見られる (実行効率で 0.6% の向上)。L3 にはオンキャッシュだが, キャッシュヒットに関連する m の要素が L2 にはすべて載らないため, 3D xymz のように性能の向上が見られな

いと考えられる。同様に 3D mxyz も性能向上が見えない。これは, 3D xyzm とは逆に初めから m の部分が L2 キャッシュに載りやすいことで性能向上が行われており, L3 オンキャッシュになっても性能向上が見えないと考えられる。

このように L3 オンキャッシュサイズまで配列サイズを変化させた結果, ThunderX2 では L3 ではなく, L2 を意識した最適化の効果があると言える。

## 5. 他計算機システムとの比較

今回評価をした ThunderX2 搭載計算機での MHD コードの性能を他の計算機システムでの MHD コードの性能と比較することで, ThunderX2 の相対的な性能や各計算機の実性能を理解することに繋がると考えられる。表 4 にこれまで MHD シミュレーションコードの性能を計測したいくつかの計算機システムの結果と ThunderX2 計算機の測定結果を示す[7, 9, 10]。今回の性能評価では 3 次元領域分割において, 4 種類の配列構造を利用したが, ITO-A の評価以外はいわゆる SoA (xyzm) と AoS (mxyz) を利用した測定しかしていない。また, CPU (GPU, コプロセッサ) 自体の性能を比較しやすいように, CPU 当たりの性能 (Rmax/CPU) を表に加えている。

ThunderX2 では, 1 次元領域分割時の性能も高かったが, プロダクトランを想定して現実的な領域分割である 3 次元領域分割で, 配列構造を AoS 形式とした場合が最も良い性能となった。4 ノード (8 CPUs) しか利用できておらず, 全体的な性能は低いため, CPU 当たりの性能を他の計算機システムと比較してみる。実行性能では, Xeon Phi KNC と同程度となっており, SKL Xeon である ITO-A より, 35 %

程度性能が低い。一方でコア数の多さもあり, Sanday Bridge, Ivy Bridge や Haswell Xeon より倍以上性能が高い。ThunderX2 では通常 1 ノード 2 ソケットのため, 2 ソケットでの性能を考えると, SX-ACE より 50%程度性能が高く, FX100 と同程度のノード性能となる。

実行効率を見ると, FX100 や Ivy Bridge までの Xeon と同程度であり, 近年の 512bit 幅 SIMD 搭載機 (Xeon Phi KNL, SKL Xeon) と比べて高い実行性能となっている。MHD コードは高 SIMD 幅計算機ではうまく性能が出ていない。配列並びでは, ベクトル計算最適化が重要な計算機では, いわゆる SoA 形式の性能が高く, RISC 系などではキャッシュヒット率が高くなる AoS 形式の性能が高くなる傾向が見て取れる。

今回の結果を他のアプリケーションの結果と比較するため, ThunderX2 におけるミニアプリの性能評価[11]と比べてみる。この性能評価では, HPCI でまとめられたミニアプリケーション群を利用して, ThunderX2 の性能評価と SKL Xeon との性能比較をしている。実行環境がアプリにより様々であり, ハイブリッド並列実行も多いため, 絶対的な比較はできないが, 計算性能 (Rmax) は SKL Xeon が高く, 実行効率は ThunderX2 が高いという点では一致している。性能の比率は, アプリケーションにより異なっているが, MHD コードと近い非定常非圧縮性熱流体シミュレーションでは, SKL Xeon が 43~57% 計算性能が高い結果となっており, こちらの差 (35%) より少し大きな性能差となっているが, ThunderX2 はおおよそ SKL Xeon の 50~65% の性能となることが想像される。

## 6. まとめ

京都大学に試験的に導入された ThunderX2 搭載計算機に対して, 宇宙プラズマを解く MHD シミュレーションコードの性能測定を行った。3 種類の領域分割を行った結果は, 3 次元領域分割がパック/アンパックに係わる時間により, 性能が高くなかった。コードの構成上可能であれば, 1 次元または 2 次元領域分割を利用することが推奨される。次に異なる配列の並びによる MHD コードの ThunderX2 での性能を評価した。ここでは, ベクトル計算向け SoA 形式よりキャッシュヒット率の高い AoS 形式の配列形状を利用することで高い性能が出ることが示された。更に, キャッシュの効果を調べるために, 配列サイズを小さくして, 性能を評価したところ, L3 にオンキャッシュになることでの性能向上ではなく, L2 キャッシュを効率的に利用できることで, 計算性能が向上することが分かった。

今回の評価結果を, これまでに性能評価した計算機システムと比較したところ, ThunderX2 単体 (1CPU) の性能は, SKL Xeon の 70% 程度, FX100 の 50% 程度の性能となっていた。1 ノード (2 CPUs) での考えると, FX100 と同等,

SX-ACE の 1.5 倍の計算性能となる。ThunderX2 は SIMD 幅が最新の Xeon などと比べて低いが, 実行効率が比較的高いため, 理論性能の差よりも, 実計算性能の差が少なくなっている。ThunderX2 に比べ, 富岳で採用される ARM 系 CPU である A64FX では, SIMD 幅も大きく (512bit), メモリバンド幅も高いため, 今回の評価結果と一概には比べられないが, 今回の結果は, ARM 系 CPU の HPC へ適用は高い性能が期待できることを示唆している。

## 参考文献

- [1] Marvell ThunderX2 (<https://www.marvell.com/server-processors/>)
- [2] Geosciences and Climate on Marvell ThunderX2 (White paper), Server Processor Business Unit Marvell, 2019.
- [3] Ogino, T, R. J. Walker, M. Ashour-Abdalla, "A global magnetohydrodynamic simulation of the magnetopause when the interplanetary magnetic field is northward", IEEE Trans. Plasma Sci. vol. 20, 1992, 817-828.
- [4] F. F. Chen, 1974. Introduction to Plasma Physics. Plenum Press, NY.
- [5] NICT Space Weather Web ページ (<http://swc.nict.go.jp/>)
- [6] Fukazawa, K., T. Ogino, and R. J. Walker (2012), "A Magnetohydrodynamic Simulation Study of Kronian Field-Aligned Currents and Aurora", J. Geophys. Res., 117, A02214, doi:10.1029/2011JA016945.
- [7] Fukazawa, K., T. Nanri and T. Umeda, "Performance Measurements of MHD Simulation for Planetary Magnetosphere on Peta-Scale Computer FX10", Parallel Computing: Accelerating Computational Science and Engineering (CSE), Advances in Parallel Computing 25, pp.387-394, IOS Press, 2014. (DOI: 10.3233/978-1-61499-381-0-387)
- [8] Arm Allinea Studio Web site (<https://developer.arm.com/tools-and-software/server-and-hpc/arm-architecture-tools/arm-allinea-studio>).
- [9] Fukazawa, K., T. Soga, T. Umeda, T. Nanri, Performance Evaluation and Optimization of MagnetoHydroDynamic Simulation for Planetary Magnetosphere with Xeon Phi KNL, Parallel Computing is Everywhere: Accelerating Computational Science and Engineering (CSE), Advances in Parallel Computing, 178 - 187, DOI:10.3233/978-1-61499-843-3-178, 2018.
- [10] 深沢圭一郎, 南里豪志, 本田宏明, スーパーコンピュータシステム ITO における MHD シミュレーションコードの計算性能・消費電力評価, 情報処理学会研究報告, 2018-HPC-166(4), 1-6, 2018.
- [11] 辻 美和子, Jean-christophe Weill, Jean-philippe Nomine, 佐藤三久, ThunderX2 Arm プロセッサにおける Fiber ミニアプリスイートの性能評価, 情報処理学会研究報告, 2019-HPC-171(4), 1-8, 2019.