April 26, 2019

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

深沢 圭一郎[1], 南里 豪志[2]

1. 京都大学学術情報メディアセンター
2. 九州大学情報基盤研究開発センター

Background 1

スーパーコンピュータの変遷

- 初期ではCRAY-1から始まるベクトル型計算機が主であったが、2000年頃からスカラ型CPUの性能向上とノード間接続・並列計算技術の向上伴い、徐々にスカラ型CPUを搭載した大並列スーパーコンピュータが増えてきた。
- 近年では，ほぼすべてのスカラ型計算機がx86型のCPUを利用しているが、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い。
- そのため、コア数の増加やSIMDなどによる同時演算数の増加により、CPU性能の向上を達成している。
- このようなCPUを利用した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どのような性能になるのかは、予測が難しい。

Background 2

スパコンの性能と消費電力

- エクサ級スパコンを開発する上で、システム消費電力が問題となっており、そこで利用可能な電力は20 MW程度（50 Flops/W）と予測されている。
- 現在のGreen500のTop 1と比べても3.9倍程度の電力性能が必要。
- 一方で、大学などの計算機センターにとっても消費電力の増大に伴い電源容量が限界に達し、また電力料金が運用コストの大部分を占めるようになっている。
  →このように消費電力の削減は重大な問題となっている。
- これらを解決するためには、ハードウェア単体だけではなく、ミドルウェア、更にはアプリケーションレベルでの電力性能最適化が重要である。
- そこで、アプリ開発者は電力性能最適化を行うために、自分のアプリがどのような消費電力特性を持っているか理解しておく必要がある。

Motivation 1

計算性能評価

- 我々は、これまでに様々なスーパーコンピュータで惑星磁気圏を解く電磁流体（MHD）コードを用いて実利用上での性能評価を行ってきた。
- 今回は、Skylake世代のXeonを採用した九州大学情報基盤研究開発センターのスーパーコンピュータシステムITOの計算性能評価をMHDコードを用いて行う。
- MHDコードは通常の流体コードに電磁場の効果を加えたコードであり、本性能評価の結果は流体系のコードに広く応用できる。
- また、これまでの性能評価結果と比較することで、ITOの現実的な計算性能を見積もることも可能である。

Motivation 2

消費電力評価

- 前述のようにエクサフロップス級計算機システムを作る上で消費電力が問題になっているように、近年は計算機システムの消費電力に注目が集まっている。
- 一般に、アプリケーション毎に消費電力特性が異なることから、自分の利用するアプリケーションがどのような消費電力特性を持つかを知っておくことが今後の計算機システムを利用する上で重要となると考えられる。
- そこで、本性能評価では、ITOにおいて、惑星磁気圏MHDシミュレーションコードの消費電力について評価を行う。
- 。

Simulation Model | MHD simulation

惑星磁気圏とは?

- 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。
- 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、その太陽風と惑星磁場が相互作用することで、磁気圏が形作られ、様々な現象が起きている。

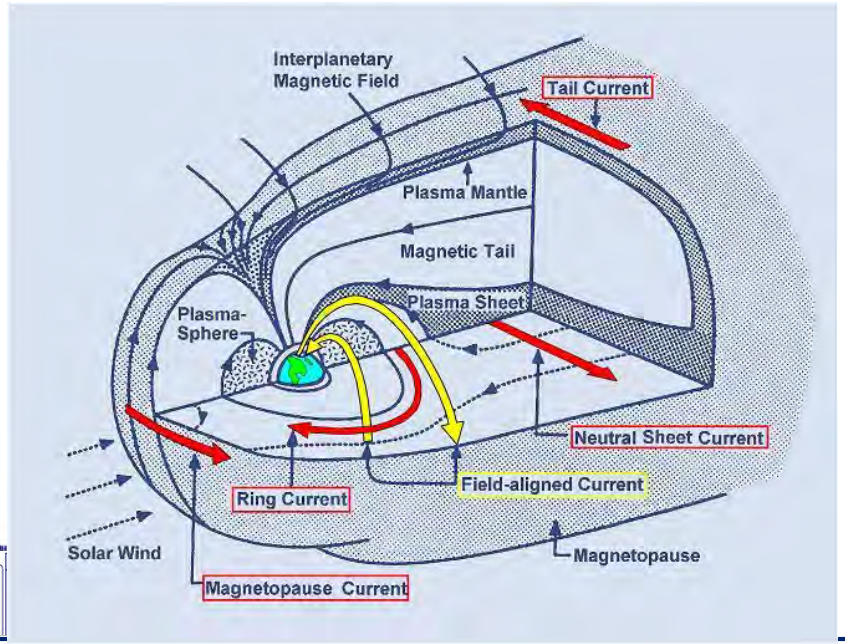

Fig.1. A sketch of the magnetosphere (modified from Kivelson and Russel [1995])

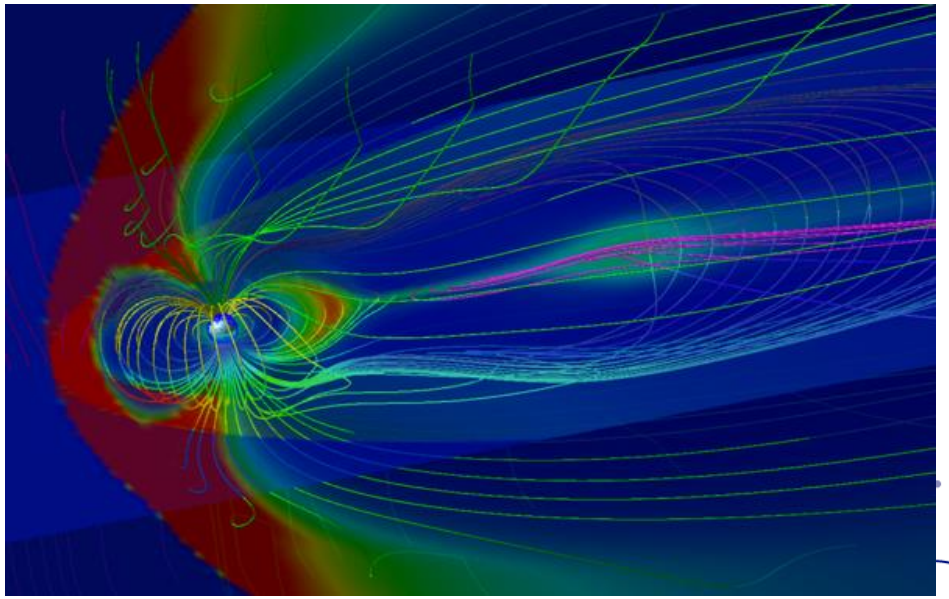

Fig. 2. MHD Simulation result of Terrestrial magnetosphere

Simulation Model | Space weather

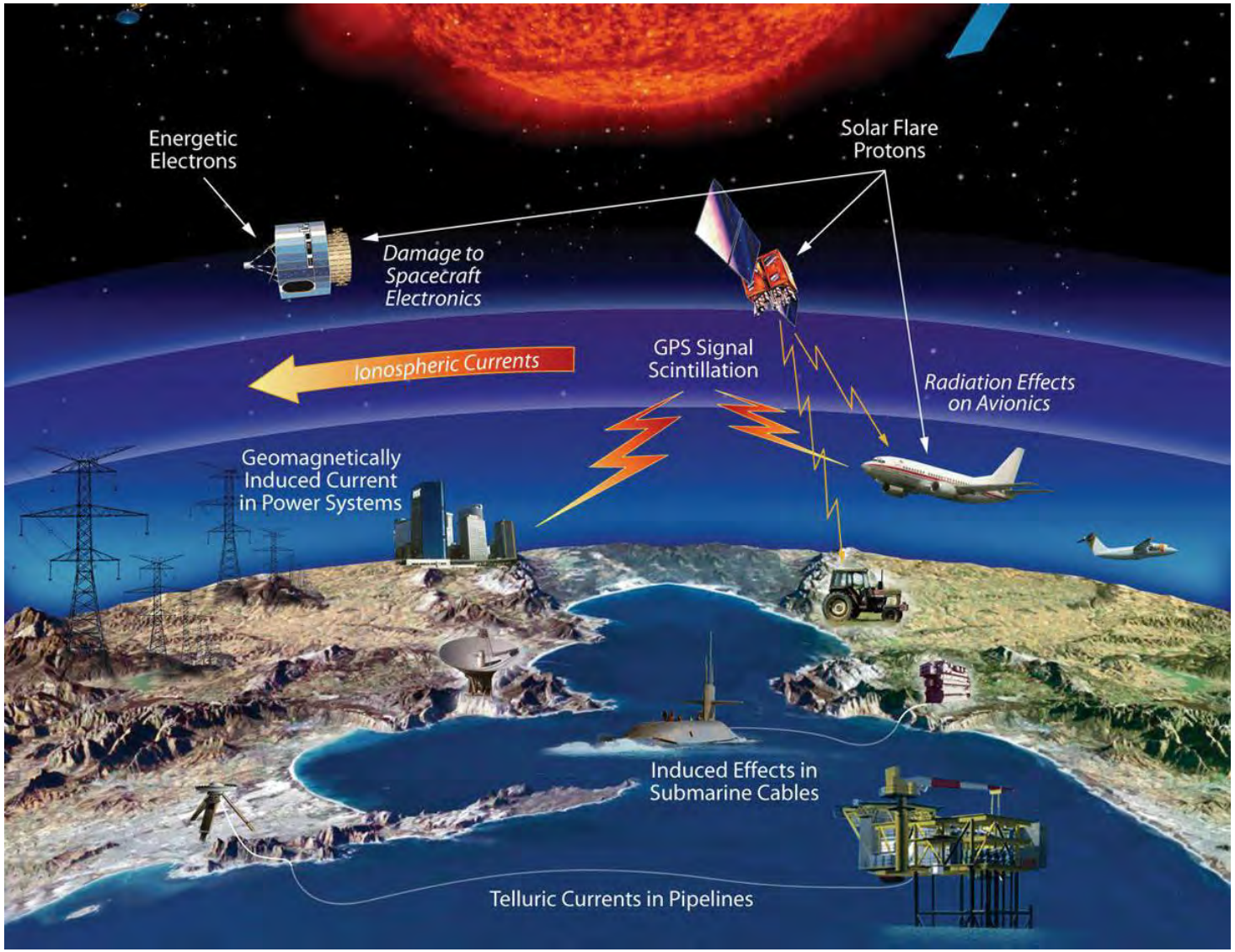

Fig. 3. Space weather affects to the environment

Simulation Model | MHD equations

Vlasov equation (collisionless Boltzmann equation)

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

$$f_s(x, y, z, v_x, v_y, v_z, t)$$

Maxwell equations

$$\begin{cases} \nabla \times \vec{B} = \mu_0 \vec{J} + \dfrac{1}{c^2}\dfrac{\partial \vec{E}}{\partial t} \\ \nabla \times \vec{E} = -\dfrac{\partial \vec{B}}{\partial t} \\ \nabla \cdot \vec{E} = \dfrac{\rho}{\varepsilon_0} \\ \nabla \cdot \vec{B} = 0 \end{cases}$$

MHD equations

$$\begin{cases} \dfrac{\partial \rho(x,y,z,t)}{\partial t} = -\nabla \cdot (\vec{v}\rho) + D\nabla^2\rho \\ \dfrac{\partial \vec{v}(x,y,z,t)}{\partial t} = -(\vec{v} \cdot \nabla)\vec{v} - \dfrac{1}{\rho}\nabla P + \dfrac{1}{\rho}\vec{J} \times \vec{B} + g + \dfrac{\boldsymbol{\Phi}}{\rho} \\ \dfrac{\partial P(x,y,z,t)}{\partial t} = -(\vec{v} \cdot \nabla)P - \gamma P \nabla \vec{v} + D_P \nabla^2 P \\ \dfrac{\partial \vec{B}(x,y,z,t)}{\partial t} = \nabla \times (\vec{v} \times \vec{B}) + \eta \nabla^2 \vec{B} \end{cases}$$

$$* \ \vec{J} = \nabla \times (\vec{B} - \vec{B}_d)$$

Simulation Model | Numerical method

Implementation

- 中心差分系のModified Leap-Frog法により、図4のような3次元8点差分を用いて、MHD方程式を差分化している。
- この計算はstaggered格子(half mesh)を用いている。
- 実際のコードの例は右のようになる。
- メモリバンド型の計算

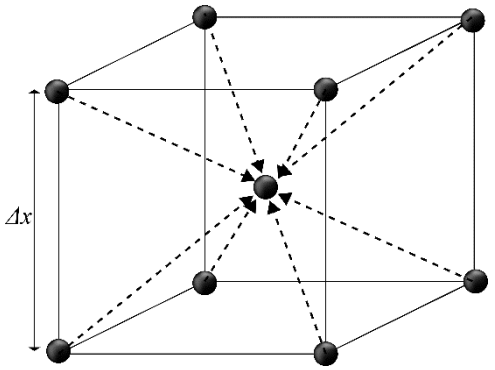

Fig. 4. Coordinate of MLF numerical method to update one value.

```
do k=1, nz
  do j=1, ny
    do i=1, nx
      u(i,j,k,8)=u(i,j,k,8)+dx*(                    &
            f(i+1,j+1,k+1,4)*f(i+1,j+1,k+1,6) &
          - f(i+1,j+1,k+1,2)*f(i+1,j+1,k+1,8) &
          + f(i+1,j,k+1,4)*f(i+1,j,k+1,6)     &
          - f(i+1,j,k+1,2)*f(i+1,j,k+1,8)     &
          + f(i+1,j+1,k,4)*f(i+1,j+1,k,6)     &
          - f(i+1,j+1,k,2)*f(i+1,j+1,k,8)     &
          + f(i+1,j,k,4)*f(i+1,j,k,6)         &
          - f(i+1,j,k,2)*f(i+1,j,k,8)         &
          - f(i,j+1,k+1,4)*f(i,j+1,k+1,6)     &
          + f(i,j+1,k+1,2)*f(i,j+1,k+1,8)     &
          - f(i,j,k+1,4)*f(i,j,k+1,6)         &
          + f(i,j,k+1,2)*f(i,j,k+1,8)         &
          - f(i,j+1,k,4)*f(i,j+1,k,6)         &
          + f(i,j+1,k,2)*f(i,j+1,k,8)         &
          - f(i,j,k,4)*f(i,j,k,6)             &
          + f(i,j,k,2)*f(i,j,k,8)  )
    end do
  end do
end do
```

Evaluation Environment

スーパーコンピュータシステムITO

- 多数のCPU計算ノードが接続されたシステムAと1ノード当たり4GPUが搭載されたシステムBにより構成される。
- 本性能評価では（主に）システムAを利用し、性能評価を行う。
- Skylake XeonはAVX-512に対応し、Goldより上では2×FMAユニット/coreであり、同時演算数は32となっている。
- メモリチャンネルがBroadwell世代の4から6と増え、バンド幅が増加している。

Table 1 Subsystem A of ITO

機種名	Fujitsu PRIMERGY CX2550/CX2560 M4	
計算ノード	CPU	Intel Xeon Gold 6154 (Skylake-SP) × 2 /node
	コア数	18 cores /CPU
	周波数	3.0 GHz (Turbo 3.7 GHz)
	理論性能	3,5 TFlops /node (倍精度)
	メモリ	DDR4 192 GB /node
	Bandwidth	255. 9 GB/s /node
	B/F	0.074
総ノード数	2,000 nodes	
総理論性能	6.91 PFlops	
ノード間接続	InfiniBand EDR 4x (100Gbps)	

Evaluation Setting of Code

並列化手法

- 1D、2D、3D領域分割
- 基本Flat MPIで、電力測定時のみhybrid MPIも利用。

配列の設定

- Normal array type : $f(i, j, k, m)$ $(xyzm)$
- 3次元領域分割時に、キャッシュヒットとベクトル化の調査のため、追加で3つの配列を利用する: $f(i, j, m, k)$ $(xymz)$、$f(i, m, j, k)$ $(xmyz)$、$f(m, i, j, k)$ $(mxyz)$

計算サイズ

- 0.5GB/process（200^3グリッド）を利用（作業配列加えると約5倍を利用）
- weak scalingで評価

Evaluation Results | Calculation performance 1

Fujitsu Fortran（FF）

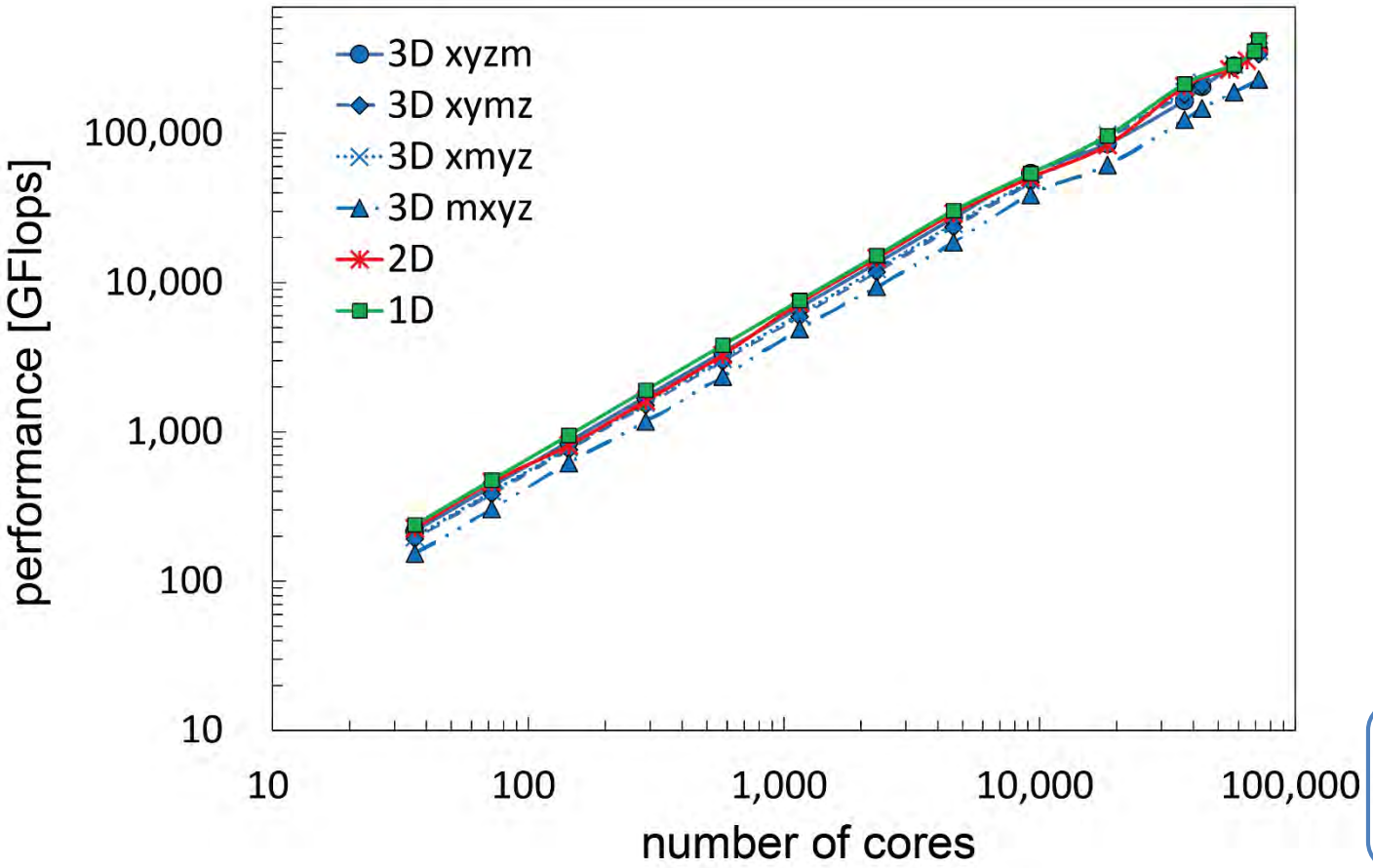

Fig. 5. Performance of MHD code with Fujitsu compiler

- これまでのXeonやベクトル型CPUと同様にベクトル長が長くなる1D、2D領域分割の性能が高くなっている。
- 2,000ノード利用時に421TFlopsの性能を達成。
- 配列順序を変えた結果は、AoS形式（3D mxyz）のみ明らかに性能が低い。
- 他は、xyzmかxmyzの性能が良く、xymzはわずかに性能が下がる。

ベクトル化とキャッシュ利用の両方を狙うには、xmyzが候補と考えられる。

富士通コンパイラはバージョン1.2.0（Nov 27 2017）＊富士通コンパイラはAVX512を使えない
-Kfast,fsimple,prefetch_indirect,prefetch_sequential,nomfunc,noparallel,simd=2

Evaluation Results | Calculation performance 2

Intel Fortran（IF）

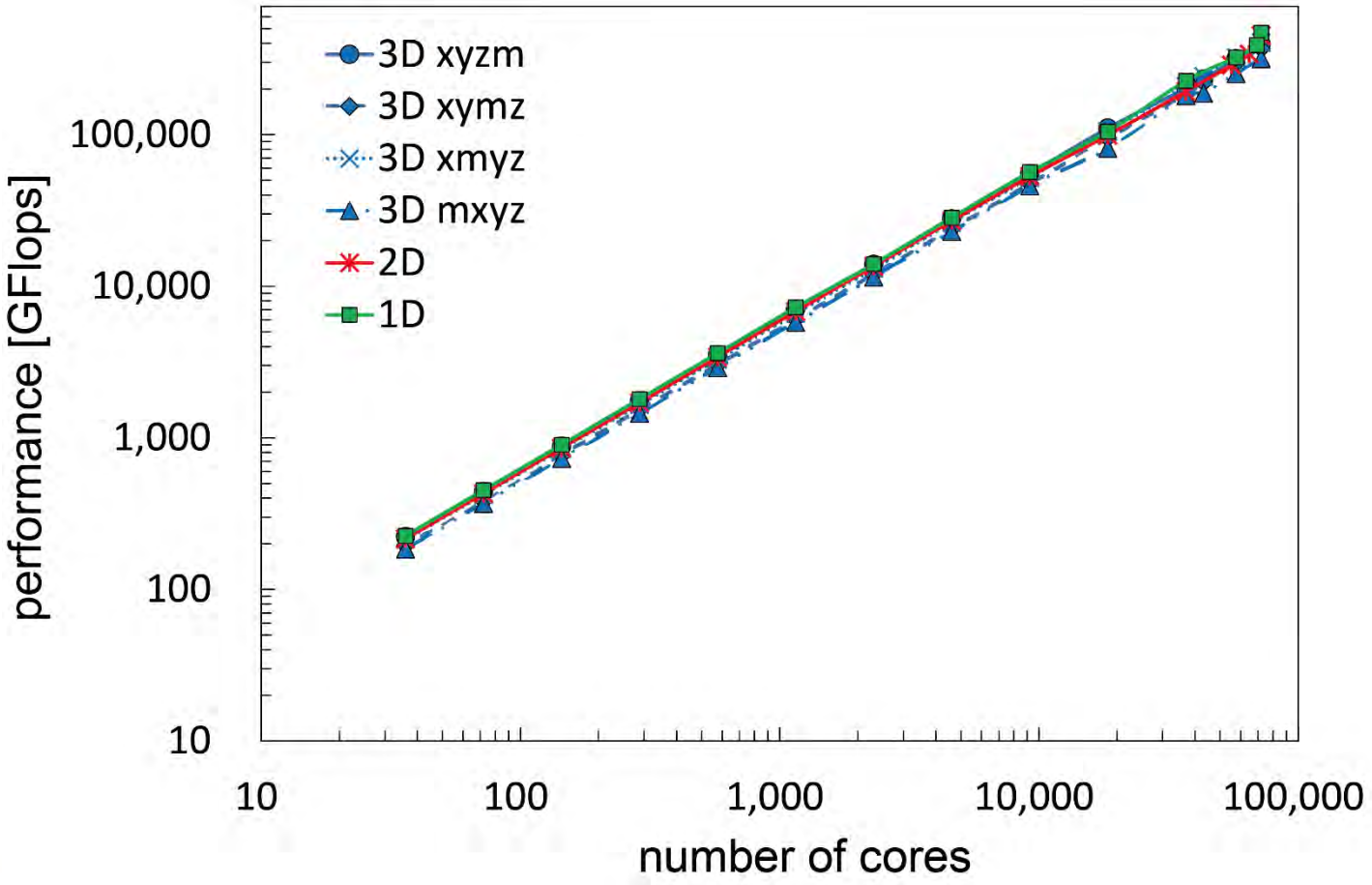

Fig. 6. Performance of MHD code with Intel compiler

- 最も高い性能は、2,000ノード利用時に1Dで約470TFlops。
  →FFより12%高い。
- 1ノードを利用した場合は、FFが5%程度高い性能（アセンブリを見るとzmmレジスタを使っていなかった）。
  →IF利用時のMPIの性能が良い。
- AoSタイプの配列構造である3D mxyzの性能がそれほど悪くない。
  →FF利用時と比べ38%高い性能。
- 配列の順序による性能の違いはFFと同じ傾向だが、その差は少なくなっている。
  →IFはより強い最適化を行っている結果と言える。

インテルコンパイラはバージョン2017＊AVX512を使うはずのコンパイラオプション
-ipo -O3 -fp-model fast=2 -xCOMMON-AVX512 -no-prec-div -no-prec-sqrt -unroll -static -align array64byte

Performance of MHD code

Performance comparison

Table 3. Performance evaluation of MHD simulation code on various computer systems.

	Core/CPU	Rpeak [TFlops]	Rmax [TFlops]	Rmax /CPU [GFlops]	Efficiency [%]	Suitable domain decomposition	CPU architecture
SX-ACE	1024/256	65.50	29.20	114.0	45	3D xyzm	Vector
K	262144/32768	4194.30	914.12	27.9	22	3D mxyz	SPARC64 VIIIfx
FX100	16384/512	576.72	91.49	178.7	17	3D xyzm	SPARC64 XIfx
CX400	23616/2952	510.11	104.23	35.3	20	3D xyzm	Xeon (SandyBridge)
HA8000	23160/1930	500.26	83.42	43.2	17	2D	Xeon (IvyBridge)
XC30	448/32	16.49	1.37	42.8	8	2D	Xeon (Haswell)
ITO-A	72000/4000	6912.00	470.10	117.5	7	1D	Xeon (Skylake)
ITO-B	72/4	5.30	0.42	104.4	8	3D xmyz	Xeon (Skylake)
Xeon Phi 5120	60/1	1.00	0.08	84.0	8	3D xyzm	Xeon Phi KNC
XC40	1088/16	48.86	4.32	273.3	9	3D xyzm	Xeon Phi KNL
Tesla K20X	2688/1	1.31	0.15	153.3	12	3D xyzm	Kepler
ITO-B GPU	3584/1	5.30	0.38	382.2	7	3D xyzm	Pascal

Power Evaluation

消費電力評価環境

- 本研究では、Inadomiらが開発したRAPL利用インターフェースであるRICを用いて、消費電力の測定、CPU消費電力に制限をかけた場合の電力性能の評価を行った。
- 計測結果にブレが大きかったため、64回の計測を行った。
- Xeon Gold 6154はベース周波数が3.0GHzで、Turbo boost（TB）時に最大3.7GHzまで上昇する。しかし、すべてのコアを利用する際はTB時には3.3GHzの周波数になり、また、AVX-512利用時には2.7GHzに周波数が下がる。
- 更にSkylake世代のCPUから、動的にCPUの周波数や動作電圧を調整する機能もあるため、実行する毎に消費電力特性が変わることが予想される。

Evaluation Results | Power performance 1

2000ノードの消費電力と性能

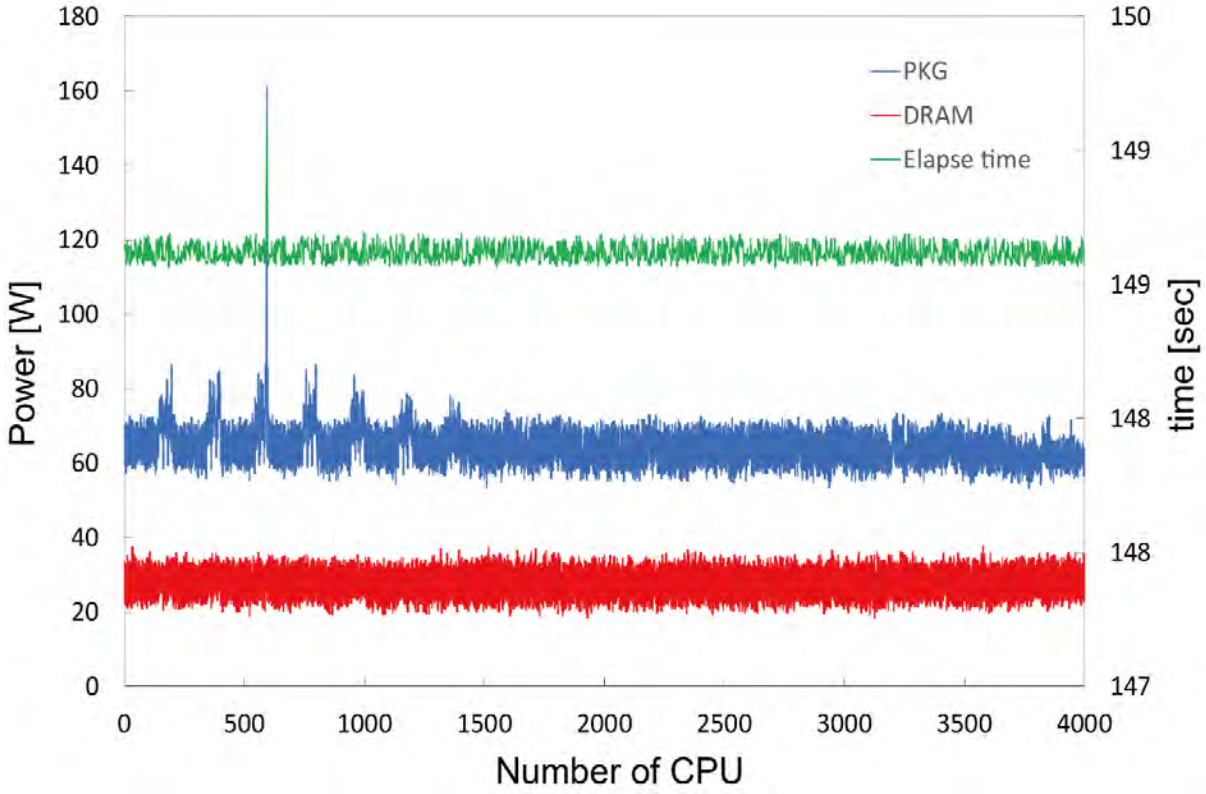

Fig. 7. Power performance of MHD code with all nodes of ITO (per CPU socket)

- CPUの消費電力は0～1,500番ではのこぎり状のブレが見えている。
- DRAMの消費電力はCPUのようなブレは見えないが、ブレの幅がCPUと同程度ある。

  →Skylake Xeonでは、CPU自体が周波数をダイナミックに変更する機能があるため、CPU消費電力も大きく変化することは分かっていたが、DRAM消費電力がこれほど大きく変化するとは想定していなかった。
- 計算時間はそれほどばらつきが見られない

  →MHDコードには集団通信が含まれていないため、各ノードの計算時間にはぶれが少ないことが分かる。

Evaluation Results | Power performance 2

2000ノードの消費電力と性能

Table 2　Power consumption characters of MHD simulation code on ITO

	経過時間 [秒]	CPU消費電力 [W]	DRAM消費電力 [W]
平均	149.556	64.639	27.762
最大	150.968	86.372	37.738
最小	148.562	52.882	18.273

- CPU消費電力の最大最小差：33.49W
- DRAM消費電力の最大最小差：19.465W

- 全ノードを利用した合計の消費電力
  CPU：258kW、DRAM：111kW、計：369kW

その他機器が100kW程度の消費電力と想定すると、MHDコードはITO全ノードで、500kW程度の消費電力と見積もられる。

- ITOのLinpack計測では1,312.8kWの消費電力となっている（4.5PFlops達成時）。
- 実アプリケーションであるMHDシコードを実行すると、その約40%の電力を消費していることとなり、実運用上と設計電力には差があることが分かる。

Evaluation Results | Power performance 3

ノード別消費電力測定結果

ノード毎、計測回毎にCPU周波数のブレが大きく、経過時間もブレている。

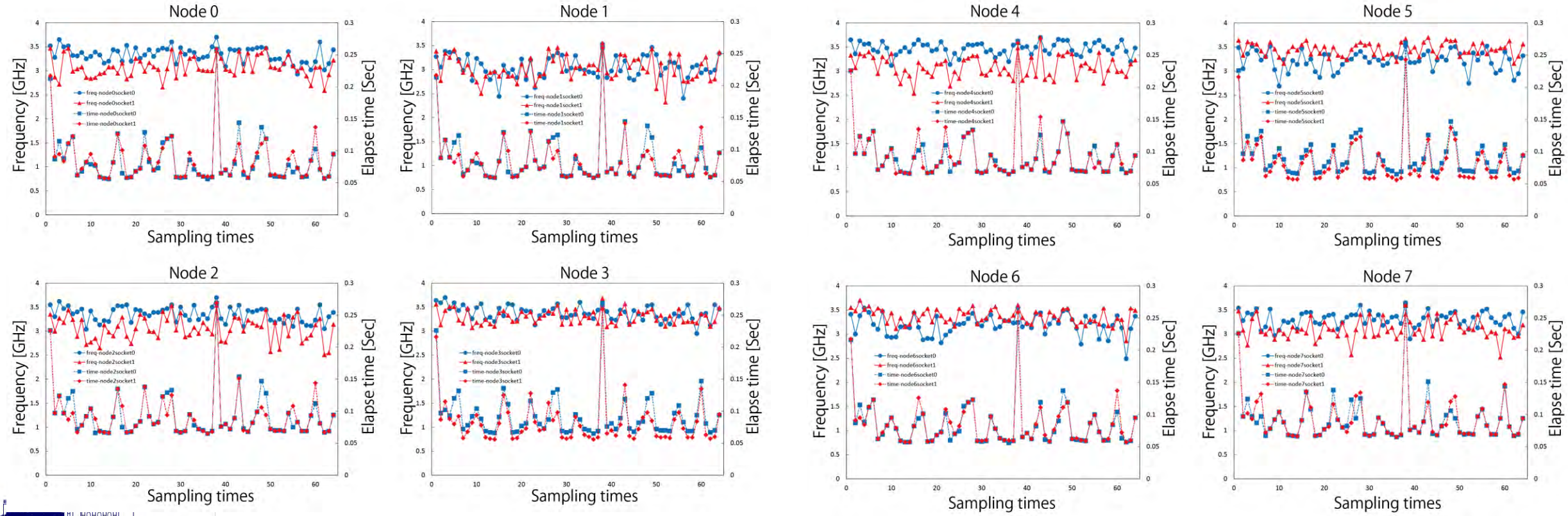

Fig. 8. CPU frequency and elapse time on each node

Evaluation Results | CPU power capping 1

CPUとDRAM消費電力変化

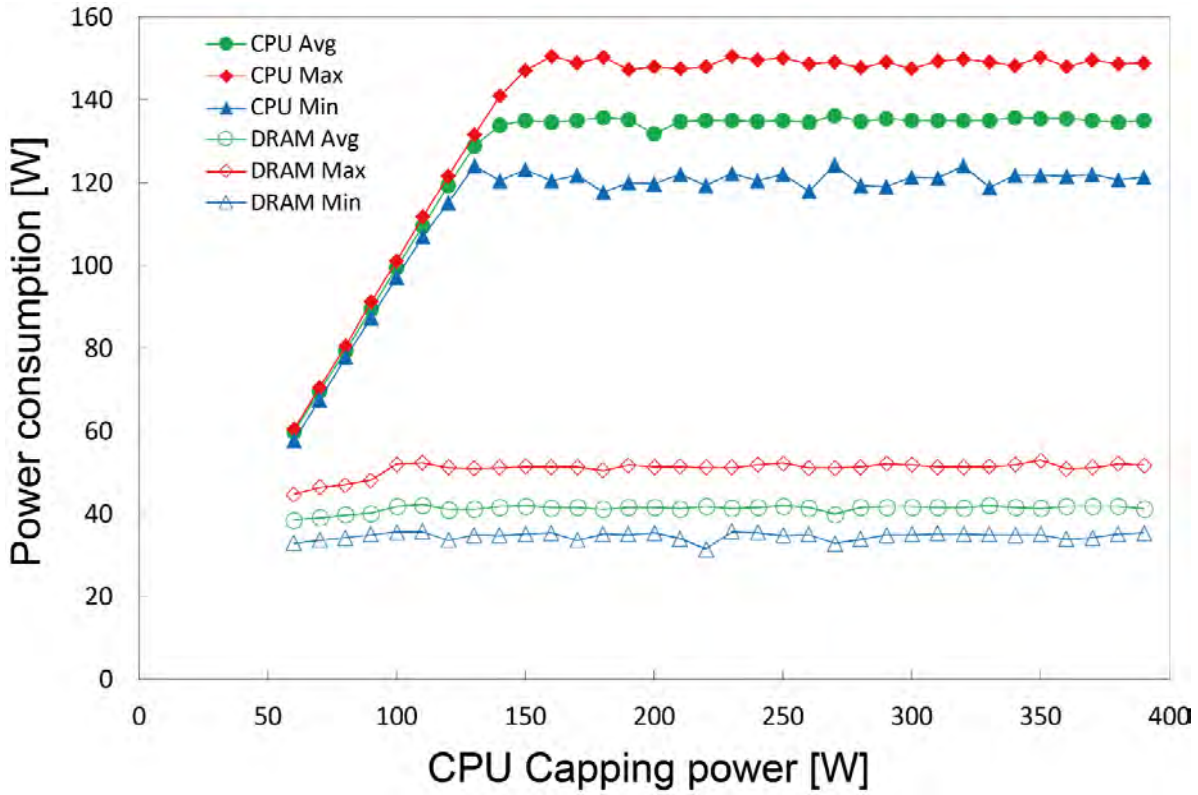

Fig. 9. Power consumption of MHD simulation code under the CPU power capping

- 少し最低CPU消費電力にバタつきが見えるが、それほど大きくはない.
- CPU消費電力に150W以下の制限がかかると、実際に消費する電力もその制限値を取るような変化を示している。
- この結果、制限が無い場合にはCPU消費電力の大きなばらつきがあったが、CPU消費電力に制限がかかると、制限値に律され、ブレが見えなくなる.
- 一方でDRAMの消費電力は、CPU消費電力制限下においてもほとんど変化がない。
- 最大DRAM消費電力だけが、少し減少しているようにも見える。

Evaluation Results | CPU power capping 2

CPU周波数と計算性能変化

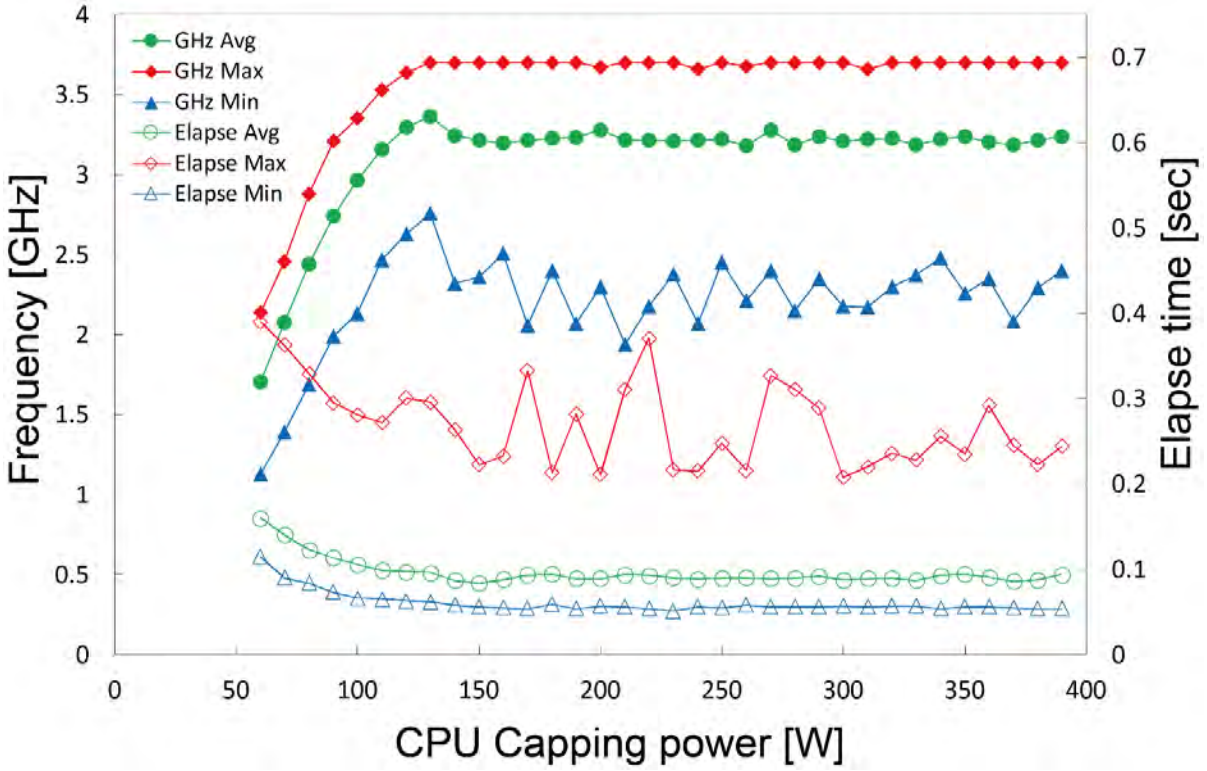

Fig. 10. Elapse time of MHD simulation and CPU frequency under CPU power capping

- 消費電力では見えなくなったブレが残っており、CPU電力性能の差が出ている。
- 60 Wの制限では、最大と最小で1GHz程度の差がある一方で、消費電力の差が無いことから、低消費電力下ではCPU電力性能の差が大きく現れている。
- 最低CPU周波数は、CPU消費電力制限が効いていない間でも大きく変動している。
  →最大計算時間のぶれ
- MHDコードの実行時間は、消費電力より周波数の影響を強く受けている。
  →計算時間もCPU電力制限下でばらつく
- CPU消費電力制限下において、最大計算時間は大きく増加している一方で、最小と平均計算時間は緩やかに増加している。

Evaluation Results | CPU power capping 3

- CPU消費電力に制限をかけない場合、Skylake Xeonでは消費電力のばらつきが現れ、システム運用の面では扱いが難しいが、消費電力に制限をかけると消費電力のばらつきが消え、電力のコントロールが容易になる。
- 一方で、消費電力制限下でも周波数のばらつきは消えず、その影響で計算時間のばらつきも残り、計算機利用者にとっては良くない効果が大きい。
- 特に最低CPU周波数への影響からか、計算時間の増加が大きく見られる。
- 現実では、電源容量の問題、季節による電力需要の問題、更には災害時における電力供給の制限など、スパコンセンターとして消費電力を制限せざるを得ないことは容易に起きている。

消費電力制限下で最大の計算性能を出すことを考えることは重要である

Evaluation Results | CPU power capping 4

Power cappingによるうま味

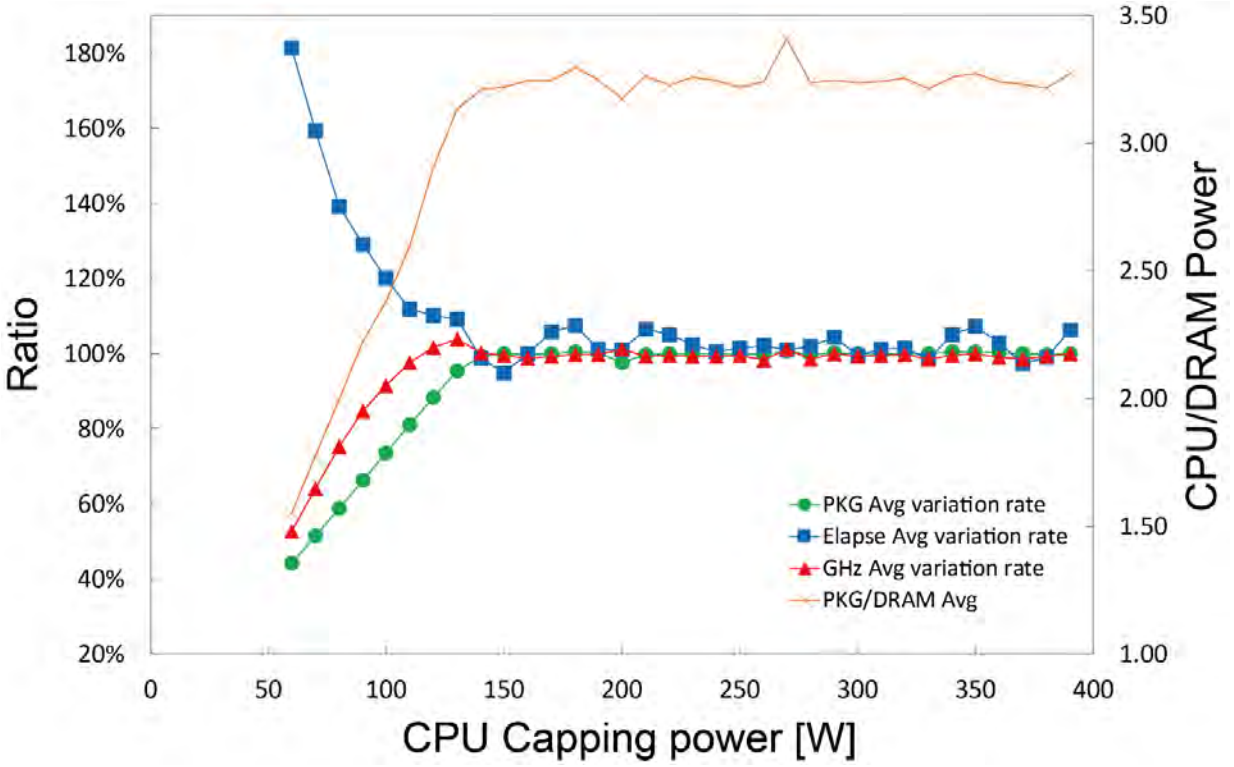

Fig. 11. Elapse time of MHD simulation and CPU frequency under CPU power capping

- CPU消費電力、周波数は消費電力制限の値に従い、ある程度一定の割合で減少しているが、130～110 W辺りでは計算時間は増加せずに一定に近くなっている。
- CPU消費電力が減少し、周波数も下がるとCPUのFlops性能が減少する。
- 一方でDRAMの消費電力は下がっておらず、DRAMのバンド幅は変化していないと考えられる。

計算機が持つB/F値が改善し、比較的高いB/Fを必要とするMHDコードでは、計算性能が下がらなかったと考えられる。

Evaluation Results | CPU power capping 5

Power cappingによるうま味2

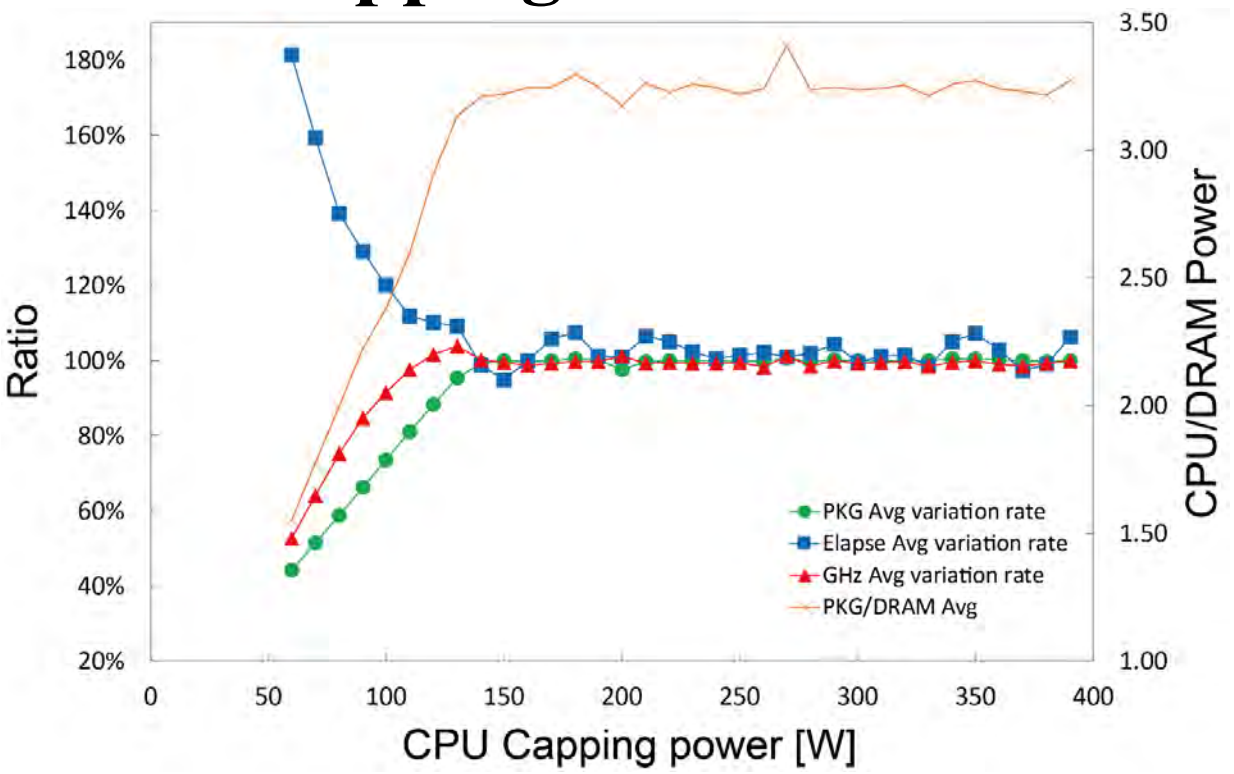

Fig. 11. Elapse time of MHD simulation and CPU frequency under CPU power capping

- 100W以上の消費電力制限をかけると計算性能は劣化しており、ある程度のCPUとDRAMの消費電力バランスが必要と想定される。
- CPU消費電力をDRAM消費電力で割った値（C/D index）を計算すると、2.6までは計算性能がそれほど下がっていない。

  →電力制限が必要な場合は、C/D Indexが2.6を下回らないような制限であれば、計算性能への影響はほとんど無く、消費電力自体は削減が可能と考えられる。
- このC/D IndexはアプリケーションのB/F値と計算機のB/F値が関連しているが、計算機システムで計算性能を調べる際に、このIndexを把握しておくことが今後重要になるかもしれない。

Summary | Calculation performance

- ✓ 九州大学の新スーパーコンピュータシステムITOを利用し、宇宙プラズマを解くMHDコードの性能測定を行った。
- ✓ 富士通とIntelコンパイラを利用した結果、単純な性能の差だけでは無く、異なる構造を持つ配列に対してそれぞれのコンパイラで最適化に差があった。
- ✓ 富士通コンパイラは、少ないノードでの性能がIntelコンパイラよりも高く、512ノード以上ではIntelコンパイラの性能が高くなり、スケーラビリティに明かに差があった。
- ✓ 今回4種類の配列構造を用いたが、ベクトル化とキャッシュ最適化の観点から考えると、単純なAoSやSoAではない構造の配列を用いることで高い性能が期待できる。
- ✓ 他システムと比べた結果、ノード当たり性能では、FX100より高く、KNLより低いが、最適化が進むと、KNL程度の性能になることが期待できる。

Summary | Power performance

- ✓ 消費電力制限を行わない場合に2,000ノードを利用した結果では、ノード毎でCPUとDRAM消費電力に大きなばらつきが見え、計算時間にはばらつきは見えなかった。
- ✓ また、計測毎に単一ノード内でのCPU周波数のばらつきが大きく、動的電力調整が強く働いていることが示された。
- ✓ 2,000ノード利用した全体のCPUとDRAMの消費電力は369kWとなり、その他を含むと500kW程度の消費電力と想定される。これはLinpack測定時消費電力の約40%となる。
- ✓ CPU消費電力に制限をかけた場合では、CPU消費電力に現れていたばらつきが無くなったが，CPU周波数や計算性能はばらつきが現れたままとなった。
- ✓ 特に最低周波数に大きなばらつきが見え、最大計算時間にもその影響が大きく出ている。
- ✓ CPU消費電力に制限をかけた場合もDRAM消費電力は余り変化が無く、これにより計算機システムのB/F値が改善し、ある制限区間では計算性能の劣化が少なくなった。
- ✓ CPU消費電力制限下でも計算性能が劣化しないことがあるため、アプリケーション毎にその条件を調べておくことが、消費電力制限がある運用では、ユーザにとって重要である。