

# CPU と DRAM の消費電力制限下での MHD シミュレーション コードの電力特性

深沢 圭一郎<sup>†1, †2, †3</sup> 青柳 睦<sup>†1</sup> 津秦 伴紀<sup>†4</sup> 吉田 匡平<sup>†4, †5</sup>  
稲富 雄一<sup>†3, †6</sup> 井上 弘士<sup>†3, †6</sup>

現在、エクサフロップス級計算機システムを実現するために、消費電力を削減することが重要な課題となっている。しかしながらアプリケーションレベルでは消費電力について研究が進んでおらず、アプリケーションが持つ消費電力特性もよく調べられていないのが現状である。そこで本研究では、CPU 消費電力、DRAM 消費電力に制限をかけた場合の惑星磁気圏 MHD シミュレーションコードがどのような消費電力特性を持つかを調べた。その結果、差分計算を行い計算負荷が高い箇所では CPU 消費電力制限の影響を強く受け、境界値設定などデータのやりとりが多い箇所では DRAM 消費電力制限の影響を受けやすいことが明らかになった。さらに CPU、DRAM の消費電力を同時に制限することで、MHD シミュレーションコードがどのような実効性能の変化を見せるかを調べ、コードにとって最適な CPU-DRAM 消費電力比率があることを明らかにした。

## Power Consumption Evaluation of an MHD Simulation under CPU and DRAM Power Capping

KEIICHIRO FUKAZAWA<sup>†1, †2</sup> MUTSUMI AOYAGI<sup>†1</sup> TOMONORI  
TSUHATA<sup>†3</sup> KYOHEI YOSHIDA<sup>†3</sup> YUICHI INADOMI<sup>†5</sup> KOJI INOUE<sup>†5</sup>

Recently to achieve the Exa-flops next generation computer system, the power consumption becomes the important issue. On the other hand, the power consumption character of application program is not so considered now. In this study we examine the power character of our Magnetohydrodynamic (MHD) simulation code for the global magnetosphere to evaluate the power consumption behavior of the simulation code under the CPU and DRAM power capping on the parallel computer system. As a result, it is confirmed that there are different power consumption parts in the MHD simulation code, which the execution performance decreases or does not change under the CPU power capping. This indicates the capability of performance optimization with the power capping.

### 1. はじめに

現在、エクサフロップス級スーパーコンピュータシステム（エクサ級スパコン）を作る上で、消費電力問題が最大の問題となっていることがよく知られている[1, 2]。エクサ級スパコンで利用可能な電力は 20 MW 程度と予測されており[2]、50 Flops/W の電力性能効率が求められている。これは現在の Green500 の Top 1 と比べても 13 倍程度の電力性能が必要となり[3]、達成が難しい状況である。一方で、計算機センターにとっても消費電力の増大に伴い電力料金が運用コストの大部分を占めるようになり、運用の面からも消費電力の削減は重大な問題となっている。これら電力問題を解決するためには、ハードウェアだけでなく、アプ

リケーションレベルでの電力性能最適化が重要なアプローチと考えられる。つまり、現在のハードウェアやアプリケーションの最適手法ではエクサ級スパコンを実現するためには不十分と考えられる。このような状況において、電力性能最適化を行うために、アプリケーション開発者は自分のアプリケーション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。

本研究では、CPU と DRAM 消費電力に制限をかけた場合、どのような電力性能特性を持つかを調べるため、電磁流体力学（MagnetoHydroDynamic: MHD）シミュレーションを実行した。MHD シミュレーションコードは流体シミュレーションコードの 1 種であり、今回の研究結果は一般の流体計算にも適用できると考えられる。評価・測定には表 1 のような特徴を持つ九州大学情報基盤研究開発センターの Fujitsu CX400 と、CX400 では DRAM 消費電力制御がうまく行えなかったために、表 2 のような自作サーバを利用した。

現在、アプリケーションの消費電力をコントロールするいくつかの手法が提案されている。例えば、Adagio Runtime [4] は DVFS (Dynamic Voltage and Frequency Scaling) を利用し、計算性能を下げずに消費電力を削減している(UMT2K や ParaDiS を利用)。近年では RAPL (Running Average Power

†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 現在、京都大学学術情報メディアセンター  
Now, Academic Center for Computing and Media Studies, Kyoto University  
†3 独立行政法人科学技術振興機構,CREST  
CREST, JST  
†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  
†5 現在、富士通  
Now, Fujitsu  
†6 九州大学大学院システム情報科学研究院情報知能工学部門  
Department of Advanced Information Technology, Faculty of Information  
Science and Electrical Engineering, Kyushu University

表 1 富士通 CX400 の諸元  
 Table 1 System of Fujitsu CX400

CPU	Architecture	8 cores Xeon E5 2680 (Sandy Bridge)
	Frequency	2.7 GHz
	Cache	L2: 256 KB/core L3 :20 MB/CPU
DRAM	DDR3-800	8 GB × 8 /CPU
B/F	51.2/172.8	0.30
Node	Number of CPUs	2
	Memory size	128 GB
System	Number of nodes	1476 (23,616 cores)
	Rmax	510.1TFlops
	Node comm.	InfiniBand FDR

表 2 自作サーバの諸元  
 Table 1 System of hand-made sever

CPU	Architecture	6 cores Xeon E5 2620 (Sandy Bridge)
	Frequency	2.0 GHz
	Cache	L2: 256 KB/core L3 :15 MB/CPU
DRAM	DDR3-1333	16 GB × 8 /CPU
B/F	85.333/192	0.44
Node	Number of CPUs	2
	Memory size	256 GB
Mother board	SuperMicro	X9DRL-iF (Chip set: Intel C602)

Limit) が Sandy Bridge 世代以降の Intel CPU で利用でき、Rountree らは RAPL を用いた電力制限下でのベンチマークアプリケーションの性能測定を行っている[5].

本研究報告の構成は以下の通りである。第 2 章では、MHD シミュレーションモデルについて説明し、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 章で消費電力制限下での MHD シミュレーションの電力測定結果を述べて、第 5 章で CPU と DRAM 消費電力割合について議論し、最後に研究のまとめをする。

## 2. MHD シミュレーションモデル

### 2.1 MHD 方程式

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。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、磁気圏という構造ができる。ここでは磁気嵐などの様々なプラズマ現象が起きている。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。

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。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。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、 $f_s(\vec{r}, \vec{v}_s, t)$  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。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。

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

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[6]. 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[7].

## 2.2 数値計算手法

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 (MLF) 法[8, 9]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1に MLF 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. また、並列化は MPI を利用し、プロセス並列だけを行っている.

電力を測定する際に、この数値計算を二つのパートに分けて評価を行った. 一つは MLF 法の計算が主である calc-part とし、もう一つを図1にあるような 1st step と 2nd step の間で行われる配列の load/store や境界値計算、MPI 通信部分を data-part とする. 計算時間の割合は、calc-part が実行時間の約 85% を占め、data-part が残りの時間となる.

## 3. MHD コードの性能評価結果

ここではMHDシミュレーションコードの性能と消費電力について測定を行った. 消費電力測定と制限についてはインテルが提供しているRAPL (Running Average Power

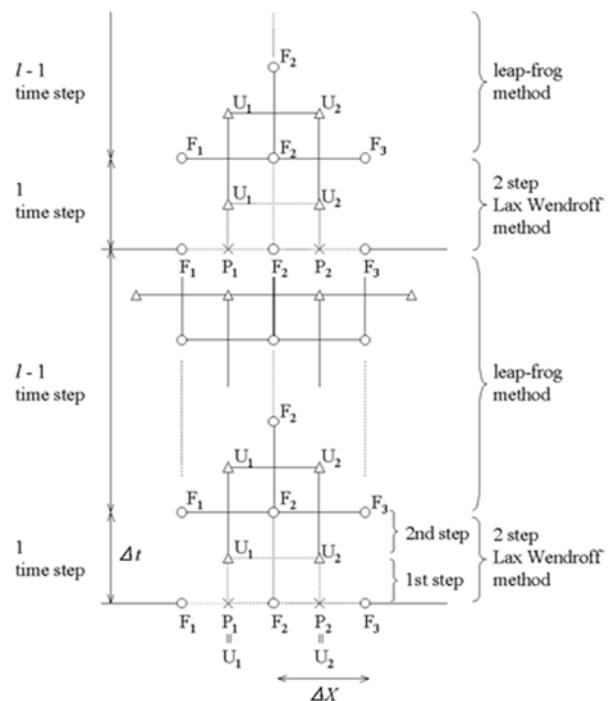


図1 Modified Leap Frog 法の計算スキーム  
 Figure 1 Scheme of Modified Leap Frog method

Limit) [10] を利用した. RAPLはCPUやDRAMなどの消費電力を測ることができ、また消費電力に制限を加えることができるインターフェースである. 本研究ではCPUとDRAMの消費電力に注目して測定を行っている.

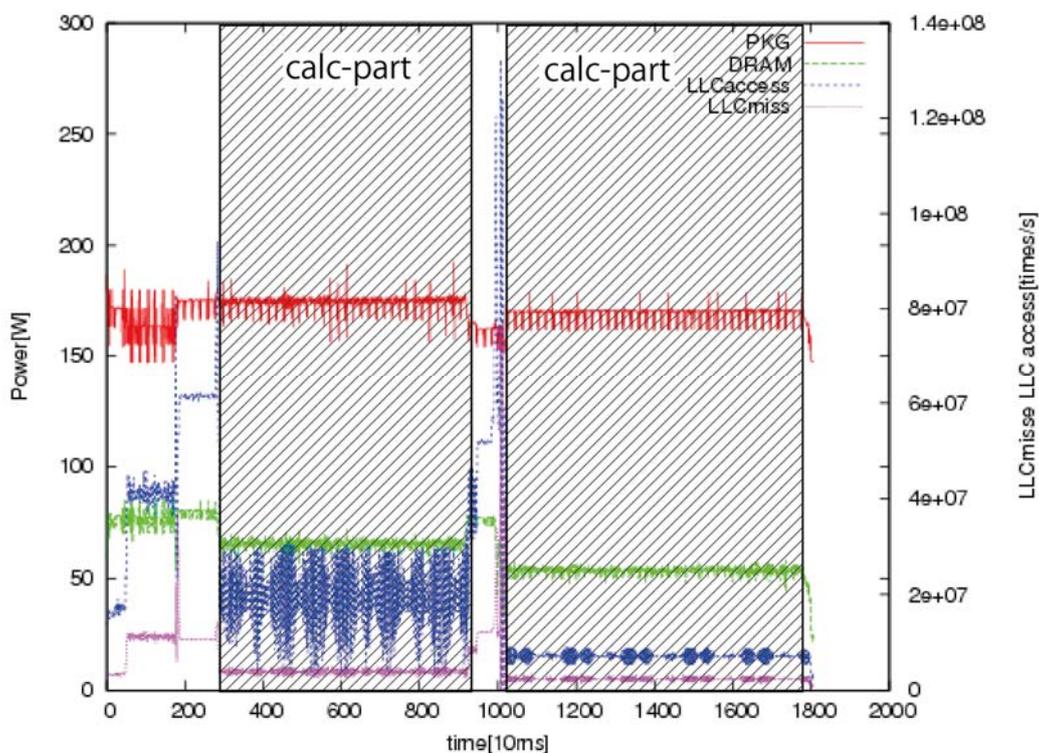


図2 RAPLで計測したPKG, DRAM, LLC access と miss rate の消費電力時間変化

Figure 2 Power consumption of PKG, DRAM, LLC access and miss rate in time line measured by RAPL

表3 MHDシミュレーションのcalc-partにおける消費電力

Table 3 Power consumption of calc-part in the MHD simulation code with several nodes

Number of nodes	CPU [W]	DRAM [W]	Elapse time [sec]	CPU/node [W]	DRAM/node [W]
1	188.32	65.05	390.58	188.32	65.05
32	5,913.20	2,106.83	394.81	184.79	65.84
64	11,985.89	4,867.42	393.68	187.28	76.05
128	23,743.39	9,723.32	390.56	185.50	75.96
256	47,485.39	19,063.96	392.37	185.49	74.47
512	95,654.56	35,059.75	393.59	186.83	68.48
1,024	187,010.42	63,209.26	388.64	182.63	61.73
1,472	272,996.83	101,796.42	392.94	185.46	69.16

表4 MHDシミュレーションのdata-partにおける消費電力

Table 4 Power consumption of data-part in the MHD simulation code with several nodes

Number of nodes	CPU [W]	DRAM [W]	Elapse time [sec]	CPU/node [W]	DRAM/node [W]
1	172.45	77.75	67.58	172.45	77.75
32	5,444.98	2,492.62	67.00	170.16	77.89
64	11,379.43	5,563.98	62.30	177.80	86.94
128	22,479.50	10,988.15	63.94	175.62	85.84
256	44,048.63	22,544.07	68.56	172.06	88.06
512	91,172.67	39,570.63	65.03	178.07	77.29
1,024	179,769.18	65,274.78	106.23	175.56	63.74
1,472	252,556.57	99,835.16	114.65	171.57	67.82

### 3.1 CPU消費電力制限が無い場合

まず、消費電力制限を加えずにMHDシミュレーションを実行し、どのような消費電力特性になるかを測定した。測定は5回行い、著しく結果が異なるものを除いて平均を取った値を測定結果とした。また本研究では複数ノードを利用する場合 weak scaling で測定を行った。この測定ではCX400を利用している。

図2に32ノードを利用しMHDシミュレーションを実行した際のCPU(PKG), DRAM, LLC access rate と LLC miss rateの時間変化を示す。この結果から、calc-part と data-part の中では大きく消費電力が時間変化することは無いことが分かる。このため、以降では各パートの平均消費電力を扱うことにする。

表3と4にcalc-part と data-part の消費電力の測定結果を示す。ここでは最大1,472ノードを利用してMHDシミュレーションの消費電力を測定した。calc-part を計算するのに390.58秒かかり、ノード当たりの消費電力はCPUが188.32W, DRAMが65.05Wとなった。data-part では Elapse time が67.58秒となり、ノード当たりのCPU消費電力が172.45W, DRAMが77.75Wとなった。前述のようにcalc-part は数値計算のコア部分を含んでおり、CPUの消費電力がdata-part と比べて9.2%も高くなっている。一方でDRAMの消費電力はdata-partの方が19.2%もcalc-partより

高くなっている。これらの傾向はコードの内容(CPUを利用する計算が主、データのload/storeが主など)をよく反映している結果と言える。

CX400に搭載されているCPUのTDP(Thermal Design Power)は260Wであり、今回の計測結果と比べてみると、MHDシミュレーションでは最大限の消費電力に達していないことがわかる。この点から消費電力が最適化の一つの目安となると提案できる。

利用ノード数を増やしていくと、表3, 4にあるように消費電力はノードに比例して増加していく。この計測では weak scaling を用いており、ノード当たりの計算量は変化しない。ノード数を変化させても、ノード当たりのcalc-part と data-part のCPU消費電力はそれほど変化していないが(~4%以内) DRAMの消費電力は変化が見える(~14%)。これがdata-partのElapse timeの変動に影響を与えている。

1,472ノードを利用した場合、calc-part と data-part の消費電力の差は20,440.26Wとなる。これは同じシミュレーションコードの中でも内容によって消費電力が大きく異なることを示しており、コードの書き方で計算性能がほとんど落ちず、消費電力は削減できる可能性を示している。

また、1,024と1,472ノード利用時にDRAMの消費電力が低くなり、Elapse timeが増えている。これはCX400のノード間接続によるもので、1,024ノードを超えたノード間は通信性能が1,024ノード間よりも低いもので構成されていることが原因と考えられる。

### 3.2 CPU消費電力制限がある場合

CPUに消費電力制限をかけるにはRAPLを利用した。CX400ではDRAMに消費電力制限をかけられなかったため、ここではCPUだけに消費電力制限をかけた1ノードでMHDシミュレーションを実行した。消費電力は102WからTDPと同じ260W(制限なし)まで制限を変化させた。

図3に計測したcalc-partのCPUとDRAM消費電力とElapse timeを示す。表3に示したようにMHDシミュレーションは185W/CPUを消費電力制限なしで使用するため、180Wまでの制限下では消費電力、Elapse timeもほぼ変化がない。消費電力制限が180Wを下回ると、その効果がよく見えてくる。制限が強くなるにつれて、Elapse timeは大きくなり、CPUの消費電力は制限通りに下がっていく。変化量を見ると、消費電力が182Wから109Wまで変化した場合(60%の変化)、Elapse timeは増加率が142%となっている。つまり電力性能割的には2%の増加と言える。消費電力制限をかけていないDRAMの消費電力はCPUの消費電力が下がるにつれて、CPUの消費電力に比べてゆっくりと下がっている。これはCPU消費電力制限によってCPUのスループットが落ち、時間当たりのメモリアクセスが減ったためと考えられる。

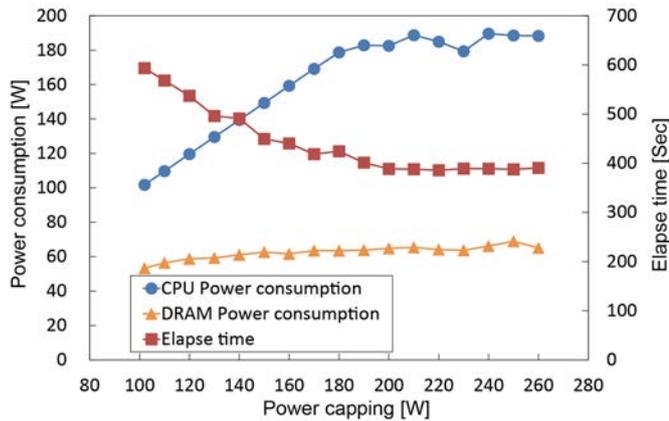


図3 CPU電力制限下での calc-part 消費電力と経過時間  
 Figure 3 Elapse time and power consumption of calc-part under the CPU power capping

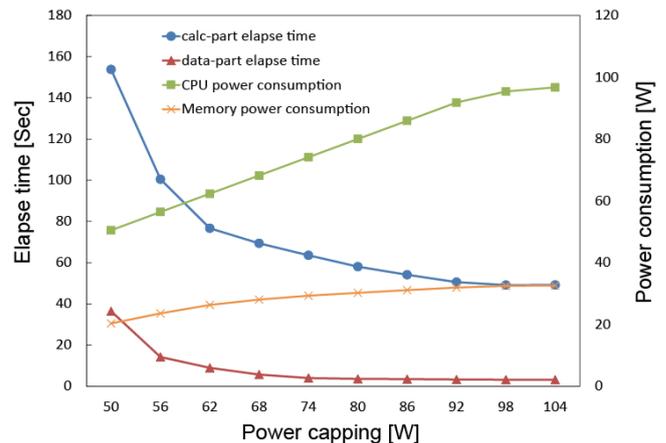


図5 自作サーバにおける CPU 電力制限下での測定結果  
 Figure 5 Measurement results under the CPU power capping on the hand-made sever

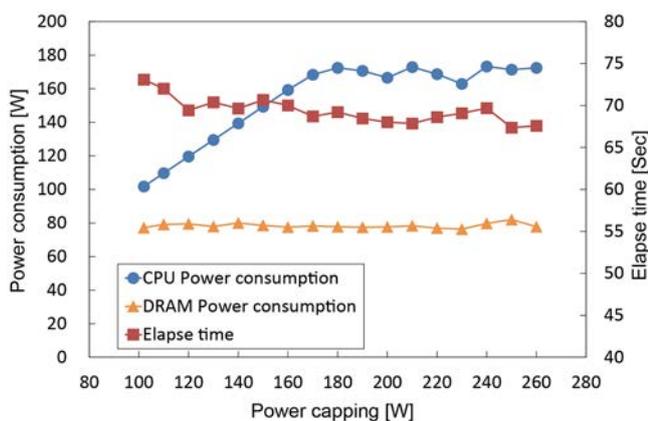


図4 CPU電力制限下での data-part 消費電力と経過時間  
 Figure 4 Elapse time and power consumption of data-part under the CPU power capping

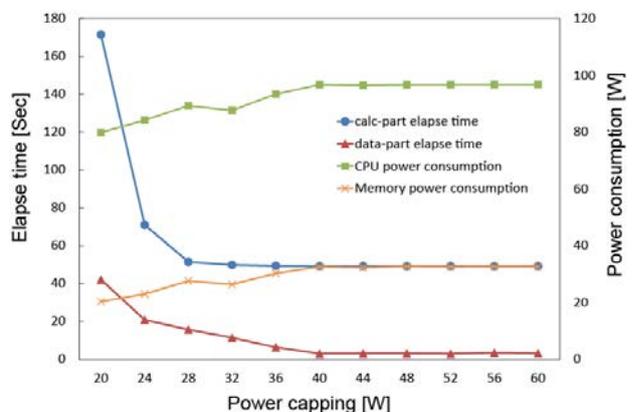


図6 自作サーバにおける DRAM 電力制限下での測定結果  
 Figure 6 Measurement results under the DRAM power capping on the hand-made sever

data-part の結果を図4に示す。消費電力制限がない場合で175W程度の消費電力だったため、data-partも180Wの消費電力制限までそれほど大きな変化は見えない。180Wの制限以下ではCPU消費電力は下がっているが、Elapse timeはそれほど変化しておらず(67~72秒)、DRAMの消費電力はElapse time以上に変化が見られない。これはデータのload/storeといったメモリを使う動作が多い部分では、CPUの消費電力を減らしても、メモリ上の動作に影響が無く、経過時間にもそれほど影響がないことを示している。

この結果はアプリケーション実行中に動的に消費電力制限を行うと、計算性能は落とさずに、消費電力を削減できる証拠の一つと言える。このような特徴を現在のアプリケーション開発者は把握していないが、これからは自分のアプリケーションがどのような消費電力特性を持つのか把握することが重要な点と考えられる。

### 3.3 CPU-DRAM 消費電力制限がある場合

CX400にはDRAMの消費電力制限がかけられなかったため、制限がかけられるSuperMicroのマザーボードを利用した自作サーバを利用して測定を行った。CX400と同様に

CPU消費電力制限がかかること、DRAM消費電力制限が機能していることを確認するために制限をCPUだけの場合、DRAMだけの場合のそれぞれを測定した。図5はCPUだけに消費電力制限をかけた結果であり、制限が強いところ以外(56, 50W以外)ではCX400の場合と同様の変化を示していることが確認できる。制限が強いところではElapse timeが極端に遅くなっており、電力不足や動作不安定が想像される。また、図6はDRAMだけに消費電力制限をかけた結果を示しており、制限無しでDRAMの消費電力は~40W程度しかないので、制限が効き始めるのが40Wより小さい制限からだが、DRAMの消費電力が下がっているのが確認できる。DRAMの場合も制限が強いところで極端にElapse timeが遅くなり、20Wでは24W時の倍以上に遅くなっている。

次にCPUとDRAMの消費電力に同時に制限をかけたどのような変化になるかを調べた。図7にcalc-part、図8にdata-partの消費電力制限下でのElapse timeを示す。calc-partのElapse timeは図5,6の結果と同様の結果となっており、

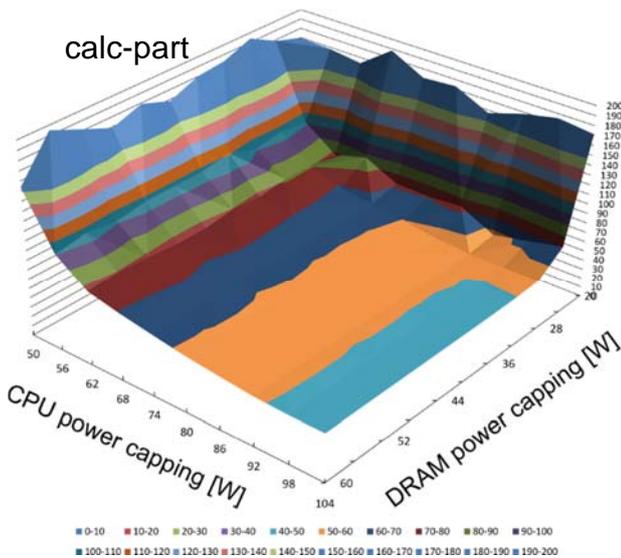


図7 CPUとDRAM電力制限下でのcalc-part経過時間  
 Figure 7 Elapse time of calc-part under both CPU and memory capping

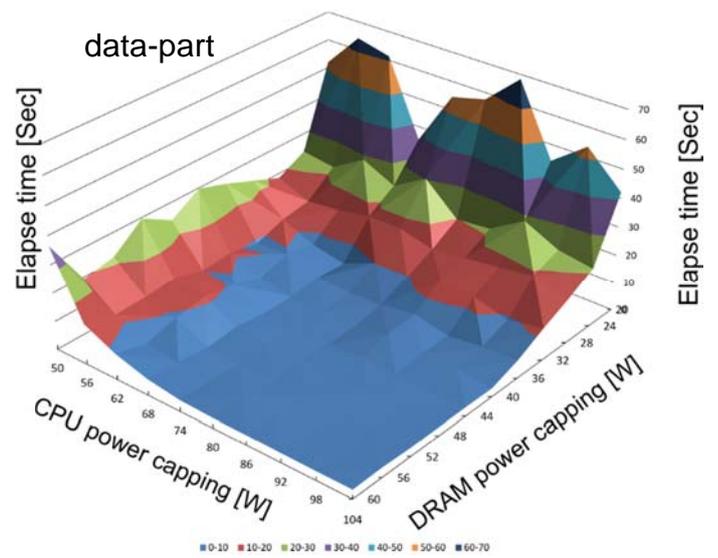


図8 CPUとDRAM電力制限下でのdata-part経過時間  
 Figure 8 Elapse time of data-part under both CPU and memory capping

CPUとDRAM消費電力制限に沿ってElapse timeは遅くなっているが、どちらかに強い制限がかかるとElapse timeは非常に遅くなる。このようにCPUだけの結果とDRAMだけの結果を足したような測定結果に見えるが、図7の奥、緑に囲まれた赤い部分などでは、周りより良い電力性能を示していると考えられる。data-partもcalc-partと同様の変化に見えるが、青から赤に変化する場所では(図8左側)、CPU消費電力を56W程度に固定して、DRAMの消費電力を下げて性能が落ちない場合があることが分かる。このようにCPUとDRAM消費電力の組み合わせ次第で、電力性能比をあげられることがある。

実際にどの消費電力の組み合わせで電力性能比が良くなるのか調べてみた。calc-partで一番Elapse timeが早かったのは、98(CPU)+60(DRAM)Wの場合で、49.10秒となっている。一方で、98+60Wで49.15秒というほぼ同じElapse timeの結果もあった。この場合DRAMの消費電力分20Wが性能を下げずに削減できている。またdata-partでは最速が104+52W時の2.94秒となっているが、92+52Wで3.00秒の結果もあった。ここではCPU消費電力分12Wが削減できている。この結果では、比較的消費電力制限が弱いときに、アプリケーションがあまり利用しない部分の消費電力を制限しても計算性能は下がらず、消費電力が削減できることが示されている。

#### 4. CPU-DRAM消費電力比

エクサ級スパコン達成の問題に電力があるように、将来的には計算で利用する電力を現在の利用ノード数指定や経過時間指定などと同様に指定することが考えられる。そこである特定の電力を利用可能な場合、CPUとDRAMに供給する電力の割合を変化させるとどのような特徴を持つの

か調査した。測定にはDRAMに消費電力制限をかけるため、自作のサーバを利用した。

図9にCPUとDRAMの消費電力割合を変化させたMHDシミュレーションの経過時間測定結果を示す。この測定結果は吉田ら[11]でのMHDシミュレーション電力性能測定と基本的に同じである。吉田ら[11]は電力分配による性能向上率について議論しているが、本報告ではそれは行わず、性能が向上する条件について議論を行う。ここでは利用可能電力を90Wから140Wまで変化させ、その中でCPUとDRAMが使える電力割合を変化させている。図で○が付いている割合はシステムがもともと持っていたCPUとDRAMの消費電力の割合を維持した場合の結果となっている(Naive)。☆が付いた割合はその利用可能電力の中で一番Elapse timeが小さくなった、最も性能が良い場合の結果となる(Best)。図から今までの自作サーバの電力制限下の挙動と同様にCPUとDRAMどちらかに極端に強い制限がかかるとElapse timeは大幅に増えている。NaiveとBestはどの場合でも一致することは無く、MHDシミュレーションではどの場合でもNaiveではCPUに電力を与えずと言える。利用可能な合計電力が大きい場合はDRAMに40Wを与える電力割合がBestとなっているが、合計電力が小さくなってくると、DRAMに電力を多く与えると性能が良くなるのでは無く、最も低い合計電力の場合でDRAM消費電力が28Wの時にBestとなっている。

ここで、Bestの場合のCPUの消費電力とDRAM消費電力の比率を計算してみると、表5のようになる。Naiveの場合は約3程度になっているがBestの場合は2.00~2.50となっている。140Wと130WではDRAM消費電力をこれ以上上げられないために少し高い比率になっていると考えられるが、2.00~2.2程度が最適なCPUとDRAMの消費電

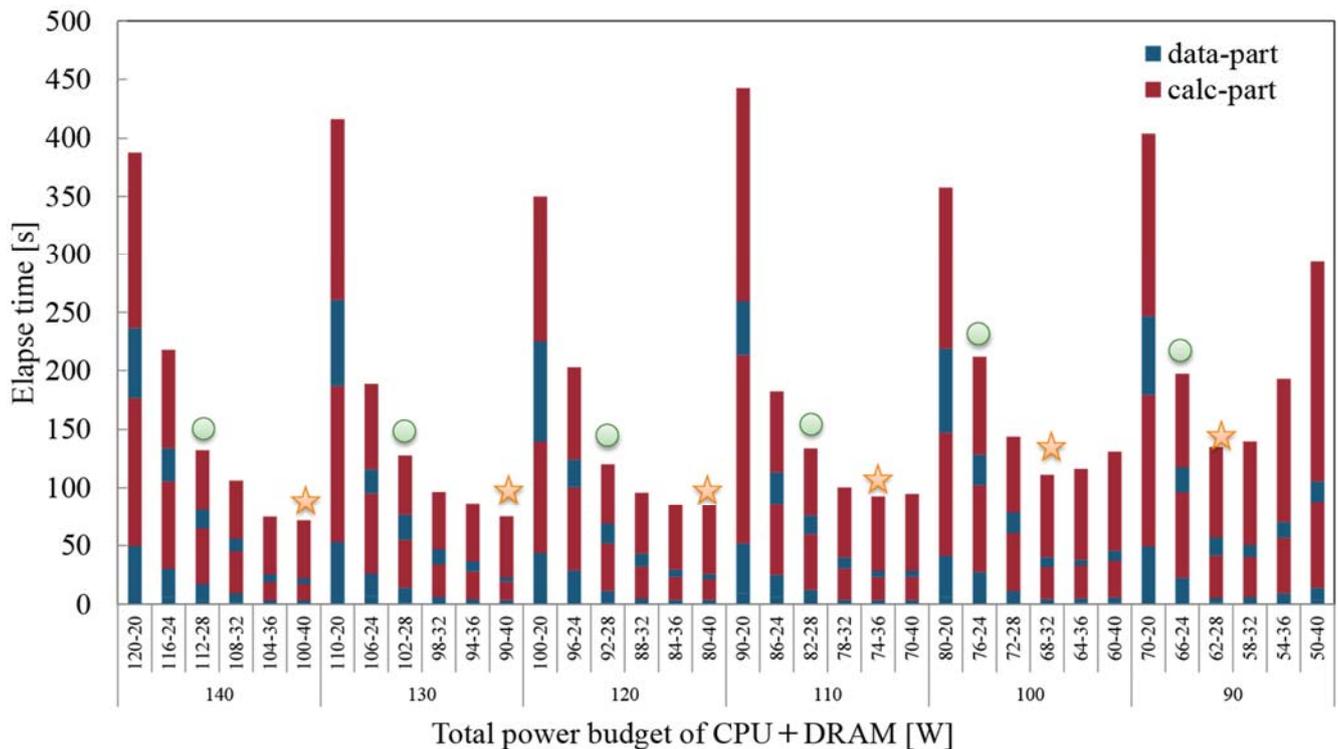


図9 一定供給電力内でCPUとDRAMの消費電力割合を変化させた場合のMHDシミュレーション経過時間  
 Figure 9 Elapse time of MHD simulation code in changing the ratio of CPU and DRAM power consumption

力比と想定される。この比率はどの計算システムであるアプリケーションはどのような割合でCPUとDRAMに電力をあたえると性能が高いのを知る指数となり得る。アプリケーションのB/F値と異なり、計算機システムに依存する指数となるため、アプリケーション単体で計算することはできないが、計算機センター等での運用において性能を下げずに消費電力を下げるための一つの指数として利用できる。

## 5. まとめ

この研究では惑星磁気圏を解くMHDシミュレーションコードが消費電力制限下でどのような電力特性を持つのか調べた。電力制限が無い場合、MHDシミュレーションコードはCPUの最大電力を使っておらず、性能最適化の余地があることが分かった。CPUに消費電力制限をかけた場合、CPU消費電力が下がり、CPU負荷が高いコードの箇所ではElapse timeも増えていったが、データのload/storeなどが主な箇所では、CPU消費電力は下がるが、Elapse timeはそれほど変化が無く、コードの内容によって消費電力を制限し

ても性能が落ちないことがあると分かった。

DRAMに消費電力制限をかけた場合、CPU負荷が高いところに比べて、データのload/storeが主なところの性能が悪くなりやすい結果となり、コードの内容を反映した挙動になることが分かった。CPUとDRAMに消費電力制限をかけた場合、比較的制限が弱い場合に、計算負荷が高いところ、load/storeが主なところで、性能を下げずにそれぞれ20W程度の消費電力削減が可能であった。

アプリケーションの最適なCPUとDRAMの消費電力比は、計算機システムによって異なるが、最適な比率があり、性能を下げずに消費電力を下げるための指数の一つとなり得る。

**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。本研究はJST、CRESTの研究領域「ポストペタスケール高性能計算に資するシステムソフトウェア技術の創出」の研究課題「ポストペタスケールシステムのための電力マネージメントフレームワークの開発」の支援を受けている。

表5 CPUとDRAM消費電力の割合

Table 5 Ratio of CPU and DRAM power consumption

Total power budget [W]	140	130	120	110	100	90
CPU/DRAM of Naïve	4.00	3.64	3.29	2.93	3.17	2.75
CPU/DRAM of Best	2.50	2.25	2.00	2.06	2.13	2.21

## 参考文献

1) P. M. Kogge, et al., Exa Scale Computing Study: Technology Challenges in Achieving Exascale Systems, in *Exascale Computing Study Report*, 2008.  
 ([http://users.ece.gatech.edu/mrichard/ExascaleComputingStudyReports/exascale\\_final\\_report\\_100208.pdf](http://users.ece.gatech.edu/mrichard/ExascaleComputingStudyReports/exascale_final_report_100208.pdf))

- 2) P. M. Kogge, and T. J. Dysart, "Using the TOP500 to trace and project technology and architecture trends," *High Performance Computing, Networking, Storage and Analysis (SC)*, 2011 International Conference for , pp.1,11, 12-18 Nov. 2011.
- 3) The Green 500 Site. (<http://www.green500.org/>)
- 4) B. Rountree, D. K. Lowenthal, B. de Supinski, M. Schulz, V. W. Freeh, and T. Bletsch, "Adagio: Making DVFS practical for complex HPC applications", Proceedings of the 23rd international conference on Supercomputing, June 08-12, 2009, Yorktown Heights, NY, USA doi:10.1145/1542275.1542340.
- 5) B. Rountree, D. Ahn, B. R. de Supinski, D. K. Lowenthal, and M. Schulz, 2012. "Beyond DVFS: A first look at performance under a hardware-enforced power bound", *Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW)*, 2012 IEEE 26th International, pp.947,953, 21-25 May 2012, doi: 10.1109/IPDPSW.2012.116.
- 6) Chang, C. L. and Lee, R. C. T.: Symbolic Logic and Mechanical Theorem Proving, Academic Press, New York (1973).
- 7) R. O. Dendy, 『Plasma Dynamics』, Oxford University Press, 1990.
- 8) 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.
- 9) Fukazawa, K., T. Ogino, and R.J. Walker, "The Configuration and Dynamics of the Jovian Magnetosphere", *J. Geophys. Res.*, 111, A10207, 2006.
- 10) Intel Quick Reference Guide to Optimization with Intel Compilers Version 13.
- 11) 吉田匡兵, 他, "CPU と主記憶への電力バジェット配分を考慮した HPC アプリケーションの性能評価", *HPC 研究会*, Vol. 2013-HPC141 No.21, 2013.