# HA8000 512 ノードを利用した電磁流体コードによる宇宙天気シミュレーション性能測定

深沢圭一郎

九州大学大学院理学研究院地球惑星科学部門

梅田隆行、荻野瀧樹

名古屋大学太陽地球環境研究所

## 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。宇宙プラズマ研究において、我々は主に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。そのため、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。宇宙プラズマの振る舞いは下記に示すようなブラソフ方程式（式 1）によって正確に記述される。

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

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、特に我々が注目している太陽風と地球の磁場の相互作用によって形成される磁気圏というグローバルな領域では、ブラソフ方程式を近似した電磁流体（MHD）近似というものがよく成り立ち、ブラソフ方程式のモーメントをとることで求められる MHD 方程式（式 2）を用いてシミュレーションをおこなっている。

$$\begin{aligned}
&\frac{\partial \rho}{\partial t} = -\nabla \cdot(\vec{v}\rho) + D\nabla^2\rho \\
&\frac{\partial \vec{v}}{\partial t} = -(\vec{v} \cdot \nabla)\vec{v} - \frac{1}{\rho}\nabla P + \frac{1}{\rho}\vec{J} \times \vec{B} + g + \frac{\mathbf{\Phi}}{\rho} \\
&\frac{\partial P}{\partial t} = -(\vec{v} \cdot \nabla)P - \gamma P\nabla\vec{v} + D_P\nabla^2 P \qquad (2) \\
&\frac{\partial \vec{B}}{\partial t} = \nabla \times (\vec{v} \times \vec{B}) + \eta\nabla^2\vec{B} \\
&* \vec{J} = \nabla \times (\vec{B} - \vec{B}_d)
\end{aligned}$$

ここで $\rho$ はプラズマの密度、$\vec{v}$ は速度、P はプラズマ圧力、$\vec{B}$ は磁場、$\vec{J}$ は電流密度、$D = D_p$ は拡散係数、$g$ は重力加速度、$\mathbf{\Phi} \equiv \mu\nabla^2\vec{v}$ は粘性、$\gamma = 5/3$ は 3 次元の比熱定数、$\eta$ は電気抵抗である。$\vec{B}_d$ は地球の固有磁場を示す。

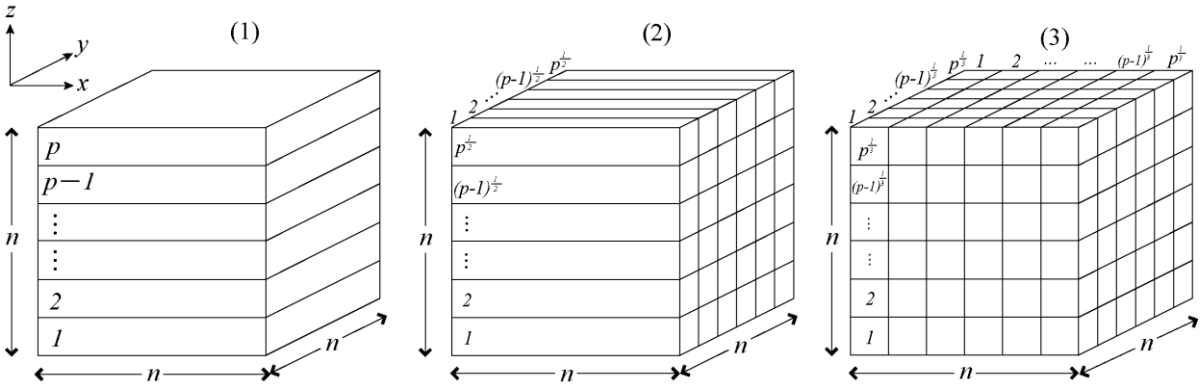
この宇宙プラズマシミュレーションは宇宙天気と呼ばれる宇宙環境の変化、擾乱の理解、予測を行う研究を支えている。宇宙空間という広大な領域を自己無撞着に扱うためにはシミュレーションが必要不可欠であり、膨大な計算機資源が必要となる。現実にこれから太陽活動が活発な時期に入るため、GPS に代表される様々な衛星や宇宙ステーションへ甚大な影響を及ぼし、不具合を引き起こす可能性が高まる。例えば 2010 年 4 月にアメリカの通信衛星 Galaxy-15（第 1 図）が静止軌道上で不具合を起こし、コントロール不能になった[1]。この原因も太陽風の変動、宇宙天気擾乱にあると指摘されている[2]。このように、宇宙環境シミュレーションを実用レベルに対応する超大規模並列計算シミュレーションの準備をしておくことは非常に重要である。そこで本 HA8000 512 ノード利用プロジェクトでは、宇宙環境（宇宙天気）シミュレーションの超並列環境における実用に向けての性能評価を行うことを目的とする。

MHD シミュレーションは近似された方程式系からなるが、中性流体に加えて電磁場を解く必要があり、メモリ負荷、計算負荷が高くなる。現在一般に行われている MHD シミュレーションでは、計算機の性能限界により、宇宙天気に影響を与えるプラズマの磁気圏界面における渦構造等を解像できないなど、解像度不足を指摘されている。例えばこれを解像するためには $0.03R_E$ ($1R_E$ は地球半径 = 6,380km)の解像度が必要と見積もられており、$4{,}096^3$ 以上のグリッドが必要となる。今回は HA8000 512 ノードにおいて最大 6,400×3,200×3,200（解像度：$0.02R_E$）の実際に今後の目標となる計算サイズの性能評価を行った。

第 1 図：打ち上げ前の Galaxy-15 衛星[3]

## ２．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 et al. [4]によって開発された Modified Leap Frog 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、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、今まで様々な計算機で性能評価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。




第 2 図: 3 種類の領域分割法。

左から 1 次元領域分割、2 次元領域分割、3 次元領域分割の概要図を示す。$n^3$ の配列を並列 p で分割している。全並列数を p としているため、2 次元領域分割では各次元で $p^{1/2}$ 並列、3 次元領域分割の場合、x, y, z 方向に $p^{1/3}$ 並列を適用している[5]。

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。領域分割には、第 2 図に示すように、1 次元、2 次元、3 次元分割が考えられ、本性能評価では最大 512 ノード、8,192 コアを利用するため、1 次元領域分割は行わず、2 次元、3 次元領域分割の評価を行う。

また、一般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 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、3 次元領域分割もこの配列定義を使った評価も行う。

今までの 1,024 コアを利用できる共同研究プロジェクトにおいて、2 次元領域分割の性能が良く、Fujitsu FX1 や Hitachi SR16000 において高い性能が出る 3 次元領域分割キャッシュチューニングコード（Type B）は、HA8000 では性能があまりでないことが分かっている[5]。しかしながら、1 万に近いコアを利用し、およそ 1 万並列の計算を行う場合では、結果が異なる可能性もあるため、2 次元領域分割、3 次元領域分割 Type A、Type B の 3 種類の性能評価を行う。

## ３．性能評価結果

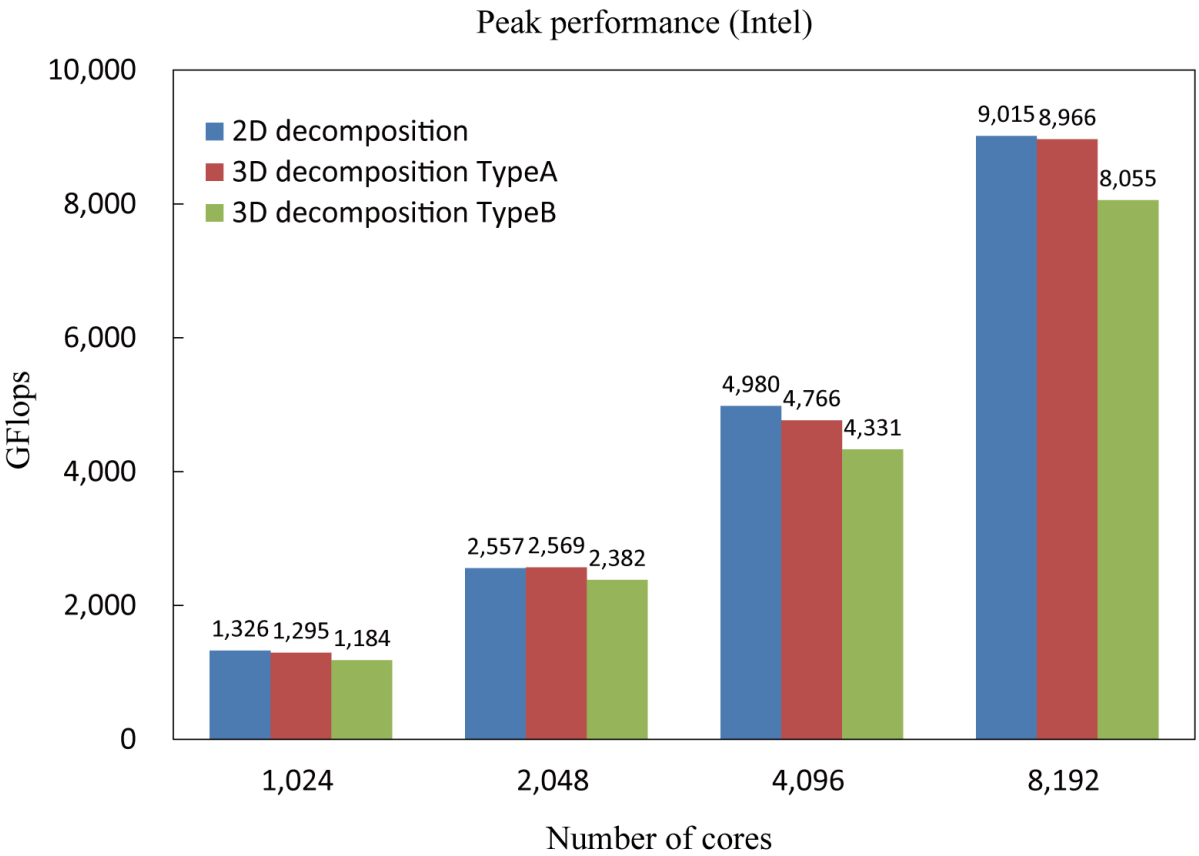
東京大学 HA8000 では 3 つの Fortran Compiler（日立製、Intel 製、PGI 製）が利用できるが、今回はその中から Intel Fortran Compiler Ver. 11.0 を主に使用した。日立最適化 Fortran Compiler は計測の時間があまりなかったため、8,192 コア利用に限り性能評価を行った。PGI 製コンパイラはうまく最適化できていないため、利用していない。コンパイラオプションとしては、日立コンパイラには、

-Oss –noparallel –autoinline=2

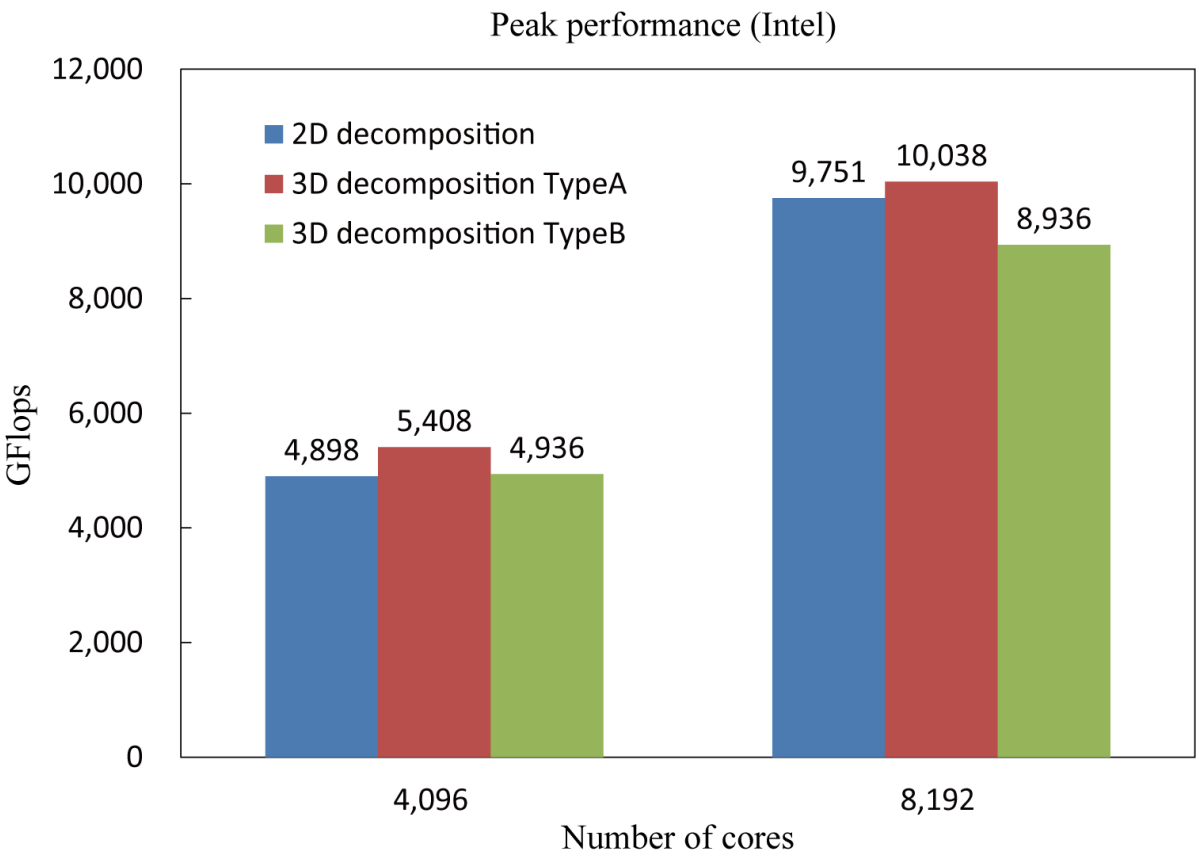
を使い、Intel コンパイラには、

-O3 –msse3 –xSSE3 –ipo

を使用した。これらからわかるように並列化はすべて MPI だけで行った（Flat MPI 並列）。ここで Intel コンパイラは SIMD Extensions (SSE)をサポートしているが、日立コンパイラはサポートしていない。




第 3 図:並列数に対する 2 次元領域分割、3 次元領域分割 Type A、Type B の実効性能。横軸が並列化数（使用 core 数）、縦が GFlops 値を表す。性能の傾向は 1,024 コアと 8,192 コア利用時で変わりはなかった。




第 4 図: 4,096、8,192 コアを利用した粒度が小さい場合の 2 次元領域分割、3 次元領域分割 Type A、Type B の実効性能。

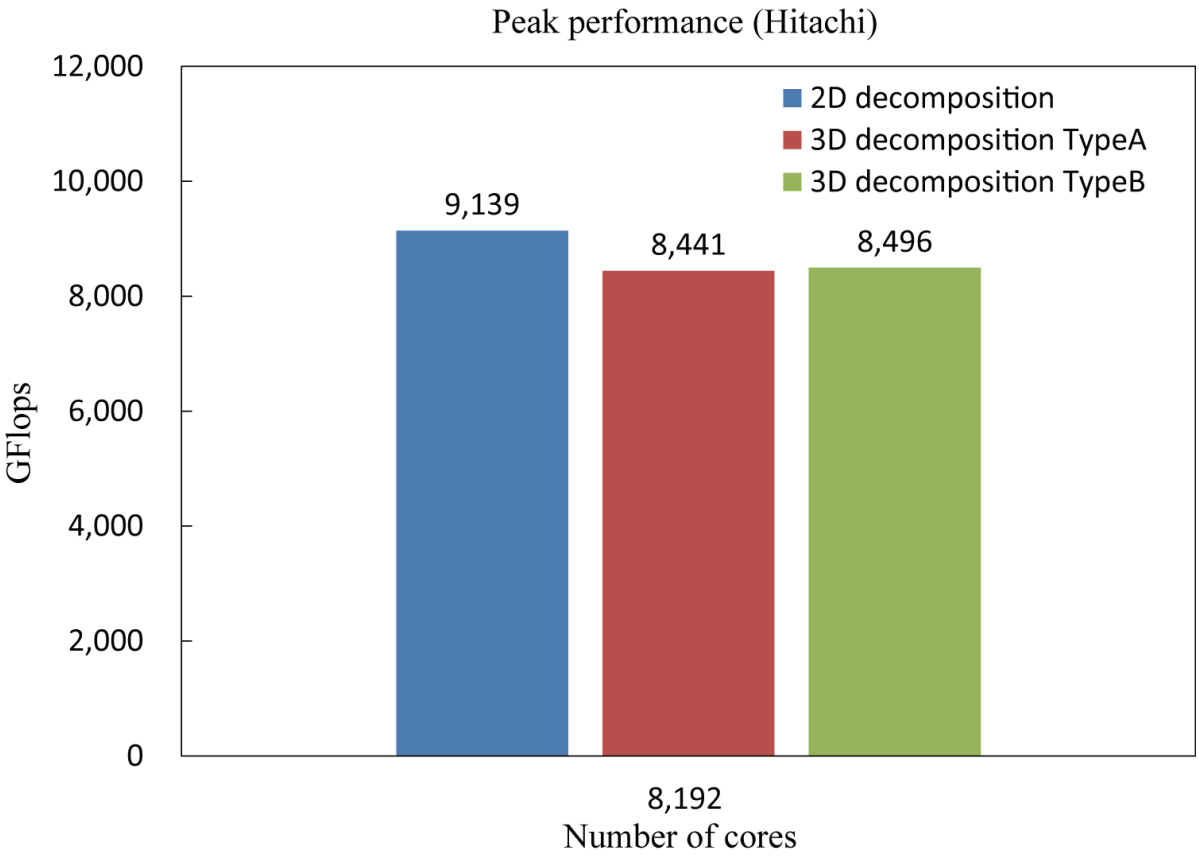
横軸が並列化数（使用 core 数）、縦が GFlops 値を表す。3 次元領域分割の性能が高くなり、最大で 10TFlops を超える性能を達成した。

並列化時の通信において、通信時間を最小限にするために、すべての境界値を入れるためのバッファー配列を用いた。また、実際の MPI で通信する際には“MPI_sendrecv”を用いた。これは送受信を一括で行う関数になり、送受信に伴うプロセスが一つで済む。例えば、MPI_send/MPI_isend”と”MPI_recv/MPI_irecv”などを使用するとそれぞれに 1 プロセスが必要となる。さらに、blocking と non-blocking の send/receive がよく大容量通信時にバッファオーバーフローするのに対して、MPI_sendrecv はほとんどのスーパーコンピュータシステムにおいて最適化されており、安定した大容量通信が行える。

まず、今までに性能評価をおこなった 1,024 コアの結果と比較するために、1 コアあたりの使用配列を固定した（weak scaling）性能評価を行った。我々が対象としている惑星磁気圏のシミュレーションではまだ解像度が足りないために、weak scaling における性能評価が研究分野として役に立つ。第 3 図に 1,024、2,048、4,096、8,192 コアを利用した 3 種類のベンチマーク結果（Intel コンパイラ）を示す。ここでは 64MB/core（1GB/node）サイズの配列を設定している。我々が使用する MHD シミュレーションでは MHD 方程式を Modified Leap Frog 法で解くためのワーク配列として 192MB/core（3GB/node）を追加で使用する。

今回の主な目的である 8,192 コアをすべて MPI だけで並列化を行い、8,192 並列の計算を行ったが、特別な問題は起きなかった。2 次元、3 次元領域分割では第 2 図に示すように、利用コ

ア数の 1/2 乗、1/3 乗の並列数が各次元に適用されるため、今回と同程度の配列利用であれば、理論上最大 $1{,}000^3$ 並列に対応が可能である。3 種類のベンチマークコードによる性能の傾向は 1,024 コア利用時と比べて、2,048～8,192 コア利用も同じであり、2 次元領域分割が良い性能を示し、次にほぼ同程度の性能を 3 次元領域分割 Type A により得た。キャッシュチューニングコードである 3 次元領域分割 Type B は少し性能が落ちる。ここでは、最大 9,015GFlops（2 次元領域分割）の性能を達成した。実行効率は、14.1%（1,024 コア）、13.6%（2,048 コア）、13.2%（4,096 コア）、12.0%（8,192 コア）となり、1,024 コア利用時に比べ、8,192 コアでは 2%程度性能が落ちている。並列化数の増加に伴う性能劣化ではあるが、我々のコードはあまり通信負荷が高くないため、ここは改善点である。




第 4 図: 日立製コンパイラによる 8,192 コアを利用した粒度が低い場合の 2 次元領域分割、3 次元領域分割 Type A、Type B の実効性能。

横軸が並列化数（使用 core 数）、縦が GFlops 値を表す。Type A と B の性能差がほぼなく、以前の結果と異なった傾向を示している。

次に粒度を変えた場合の性能評価結果を述べる。今までのベンチマークに比べてコア辺りの使用配列数を 4 分の 1 に設定し、粒度を小さくした場合の性能を調べた。第 4 図にその結果を載せる。ここでは 4,096 並列、8,192 並列の結果を載せている。粒度を小さくすることで、3 種類すべて性能が上がっていることがわかる。特に 3 次元領域分割 Type A の性能が良くなり、2 次元領域分割よりも明らかに性能が高い。ここでは最高 10,000GFlops の性能が得られた。また、実行効率では 4,096 並列で 14.4%、8,192 並列で 13.3%を達成している。粒度が大きい場合と比べて 1%程度高く、粒度が高い場合において最適化の必要があることがわかる。図には載せて

いないが、1,024 コア利用した粒度が小さい計算では 15%を超える性能を得ている。一方で粒度が通常の 2 倍程度の計算を行ったが、8,192 コアを利用し、実効性能 8,850GFlops、実行効率 11.7%とそれほど性能劣化は見られていない。

最後に日立製コンパイラを利用した結果について述べる。時間の関係上日立製コンパイラでは 8,192 コアを利用した性能測定しか行えなかった。第 5 図に 8,192 コアでの日立コンパイラの結果を載せる。利用した配列サイズは第 2 図と同様である。ここでも今までと同様の 3 種類のベンチマークを行った。性能としては Intel 製コンパイラと同様に 2 次元領域分割がもっとも良かったが、3 次元領域分割は Type A、B でほぼ差がなく、むしろ Type B の方が少し良い性能を達成した。この傾向は以前の 1,024 コアにおける性能評価では見えていない[5]。最大の実行性能としては 9,139GFlops、実行効率は 12.1%を達成した。この結果は Intel 製コンパイラの結果とほぼ同性能である。以前の 1,024 コアでの計測では明らかに Intel 製コンパイラの性能が高かったことを考えると、コンパイラのバージョンアップにより最適化度が変わっている、または並列化効率が日立製コンパイラ利用時の方が良いと考えられる[5]。そのため日立製コンパイラを利用した 8,192 コア以外、また粒度の異なった性能評価も行う必要がある。

## ４．まとめと 512 ノード利用時気づいたこと

H22 年度 HA8000 512 ノード利用サービスにおいて、電磁流体コードによる宇宙天気シミュレーション性能測定を行った。512 ノード、8,192 コアという 1 万に近いコア数を用いて並列実行を行ったが問題なく動作した。いくつかの性能評価を行えた Intel 製コンパイラの結果では、以前の 1,024 コア利用時と同様に 2 次元領域分割、3 次元領域分割 Type A の性能が高かった。キャッシュヒットを考慮した 3 次元領域分割 Type B がもっとも性能が悪かった。8,192 コアまで利用すると実行効率が weak scaling の評価でおよそ 2%下がった。また、粒度を下げた場合では、特に 3 次元領域分割 Type A の性能が良く、4,096 コア利用時に 14%、8,192 コア利用時に 13%を超える性能を達成した。日立製コンパイラを利用した場合、2 次元領域分割の性能が高いことは以前と同じであったが、3 次元領域分割 Type B の性能が Type A と変わらない結果になった。また最大性能も Intel 製コンパイラの結果とほぼ同様であり、以前とは異なった結果となった。この原因はもう少し調べる必要がある。

今回 MPI だけの並列化で実行テスト、性能評価を行った結果、まだたくさんの並列化数を MPI だけでコントロールできると感じた。理論的にはまだまだ並列化は可能だが、並列化効率の減少により、自動並列、OpenMP と MPI を組み合わせた Hybrid MPI の方が性能が出る場合も考えられる。次に 512 ノード利用の機会があればベンチマークとして Hybrid MPI の性能も測定することで、有意義なデータがとれる。

今回 512 ノード利用時に、数回に 1 回というような割合で、実行性能が非常に悪くなることがあった。リコンパイルも行わずに再度ジョブを投入すると今度は正常に動作するというような現象であった。大多数のコアを利用する場合に問題視されているのはコアの故障率が分子が増えることで上がることであり、1 つのコアに足を引っ張られる可能性が上がることである。今回の現象原因は不明だが、上記のようなことが原因かもしれない。

## 参 考 文 献


[1] Intelsat corporation [http://www.intelsat.com]

[2] 地磁気世界資料解析センター News No.121 [http://wdc.kugi.kyoto-u.ac.jp/wdc/news/1005.html]

[3] Orbital Sciences Corporation [http://www.orbital.com/]

[4] 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

[5] 深沢圭一郎、梅田隆之、荻野瀧樹、高効率並列電磁流体コードによる HA8000 クラスタシステムの性能評価、2010 年ハイパフォーマンスコンピューティングと計算科学シンポジウム論文集、133-140、2010.