Vol.2017-HPC-162 No.1
2017/12/18

# Xeon Phi KNL におけるブラソフコードの性能評価(2)

梅田 隆行[†1]　　深沢 圭一郎[†2]

Vlasov コードは宇宙空間を満たす無衝突プラズマの第一原理シミュレーション手法である．Vlasov シミュレーションでは，位置及び速度で与えられる超多次元位相空間における荷電粒子の分布関数の時間発展を，運動論方程式によりEuler 型の数値解法を用いて直接解き進めている．4 次元以上の空間を扱うシミュレーションでは，ノードあたり，あるいはコアあたりに使用できるメモリ容量の制限から，数値解法や性能チューニングにおいて様々な工夫が必要である．本研究グループはこれまでに様々な HPC 関連プロジェクトと通じて，Vlasov コードの性能チューニングを行ってきた．本講演では, Xeon Phi Knights Landing (KNL)プロセッサにおいてメモリモードとクラスタモードを変更した時のVlasov コードの性能測定結果について報告する．

# Performance evaluation of Vlasov code on the Xeon Phi KNL (2)

TAKAYUKI UMEDA[†1]　　KEIICHIRO FUKAZAWA[†2]

Vlasov code is a first-principle simulation method for collisionless space plasma. The Vlasov code solves the time development of phase-space distribution functions of charged particles in hyper-dimensions based on fully kinetic equations with the Eulerian grids. Since the distribution functions are defined in more than four dimensions, the Vlasov code requires high-resolution and high-performance numerical schemes which should work in limited computational memory per node or per core. Our Vlasov code has been made performance tuning on various scalar CPU architectures under Japanese HPC projects. In the present study, the performance of our Vlasov code is measured on the latest CPU KNL (Knights Landing) Xeon Phi with various combinations of the memory mode and the cluster mode.

## 1.　はじめに

我々が住む宇宙の 99.99%以上の体積はプラズマと呼ばれる電離気体で占められている．宇宙空間に存在するプラズマの大部分は密度が非常に小さく無衝突状態にあり，宇宙プラズマ（無衝突プラズマ）を理解することは，宇宙の本質的な理解につながる．

我々が住む地球周辺の宇宙環境は，太陽から放出された高速のプラズマ流である太陽風及び太陽風が運ぶ惑星間空間磁場（太陽の固有磁場）と，地球の固有磁場との相互作用によって複雑な磁気圏構造を形成している．プラズマ放出現象をはじめとする太陽の様々な変動により，宇宙飛行士の被曝，人工衛星の故障や通信障害に繋がる地球磁気圏・電離圏の環境変動が引き起こされ，これを宇宙天気と呼ぶ．近年の国際宇宙ステーションでの活動や人工衛星の打ち上げなど，日本においても宇宙利用が現実的になってきており，宇宙天気の予報・予測に繋がる宇宙プラズマ研究は極めて重要である．

地球磁気圏内には，プラズマの密度や温度などの物理パラメータが異なる様々な領域が生じる．その領域間の境界層で現れる不安定性（平衡状態の破れ）は，磁気圏の変動に大きな影響を与えていると考えられている．グローバル磁気圏構造に対して，境界層不安定性は中間（メゾ）スケール現象と呼ばれる．これらのグローバル及び中間スケールの現象は，粒子運動論を扱う方程式である Vlasov（無衝突 Boltzmann）方程式の 0 次・1 次・2 次のモーメントを取ることによって求められる磁気流体力学（MHD）方程式によって記述される．しかし，近年の科学衛星による高精度な「その場」観測では，中間スケールの不安定性において MHD 方程式で記述できる物理過程と粒子の運動論方程式によって記述できる物理過程が結合していることを示唆している．これらのマルチスケールの磁気圏変動である宇宙天気を真に理解するためには，全てのスケールをシームレスに扱える運動論方程式（第一原理）によるシミュレーションが本質的である．

プラズマの運動論シミュレーションには 2 つの手法がある．1 つは，プラズマ粒子であるイオンや電子などの個々の荷電粒子の運動を，Newton－Coulomb－Lorentz 方程式により解き進める PIC（Particle-In-Cell）法である．格子点（Cell）上に定義された電磁場中を粒子が動きまわることから，このように呼ばれている．宇宙空間に存在する膨大な数の荷電粒子を有限の計算機資源で扱うことは不可能であるため，ある程度まとまった数の荷電粒子の集団を 1 つの“超”粒子として扱う．PIC 法はその数値解法の完成度が高く，プラズマ科学分野では広く用いられている．しかし，プラズマを超粒子として扱うことにより熱雑音が大きくなること，電荷密度や電流密度などの荷電粒子の運動に起因する場の量を格子上に割り振る際に生じる高波数モードが数値誤差として蓄積すること，さらに並列化の際に負荷のバランス（各プロセス内の粒子数の均一性）を保った

†1 名古屋大学宇宙地球環境研究所
Institute for Space-Earth Environmental Research, Nagoya University
†2 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

めに特殊なデータの分割が必要になることなどの欠点がある．

一方もう 1 つの手法である Vlasov 法は，位置－速度位相空間に定義されたプラズマ粒子の分布関数の発展を Vlasov 方程式により直接解き進める方法である．格子点上に定義された分布関数は熱雑音を持たず，また流体シミュレーションと同様に並列計算も容易である．しかし，Vlasov 方程式は実空間 3 次元及び速度空間 3 次元の計 6 次元を扱う方程式であり，コンピュータで解くには膨大なリソースを必要とする．このため，その手法の開発はあまり進んでいない．実際，ここ数年の HPC プロジェクトによる計算機環境の飛躍的に向上によって手法の開発が進み，実空間 2 次元及び速度空間 3 次元の 5 次元シミュレーションがようやく実用の域に達しつつある段階である．

本研究の最終的な目的は，プラズマシミュレーションとしては「次々々」世代の技術にあたる第一原理 Vlasov シミュレーション手法を世界に先駆けて確立し，プラズマ科学に基づいた宇宙天気の実現に貢献することにある．そのための準備として，現存する超並列計算機上のおける 5 次元 Vlasov コードの性能評価及び性能チューニングを行っている．

これまでの研究において様々な超並列計算機での Vlasov コードの性能評価を行ってきた．本研究では，最新の CPU である KNL(Knights Landing) Xeon Phi における Vlasov コードの性能測定を行う．また，これまでの Xeon プロセッサにおける性能との比較を行った．

## 2. 計算手法の概要

### 2.1 基礎方程式

無衝突プラズマの振る舞いは，以下の Vlasov（外力を電磁力とした無衝突 Boltzmann）方程式によって記述される．

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

ここで $\vec{E}$，$\vec{B}$，$\vec{r}$ と $\vec{v}$ はそれぞれ電場，磁場，位置，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プラズマ粒子の分布関数であり，$s$ はイオンや電子など種類を示す．$q_s$ と $m_s$ はそれぞれ電荷と質量を表す．

プラズマ粒子の分布関数は，電磁場によって変形する．電磁場の時空間発展は以下の Maxwell 方程式によって記述される．

$$\nabla\times\vec{B}=\mu_0\vec{J}+\frac{1}{c^2}\frac{\partial\vec{E}}{\partial t} \tag{2.1}$$

$$\nabla\times\vec{E}=-\frac{\partial\vec{B}}{\partial t} \tag{2.2}$$

$$\nabla\cdot\vec{E}=\frac{\rho}{\varepsilon_0} \tag{2.3}$$

$$\nabla\cdot\vec{B}=0 \tag{2.4}$$

ここで $\vec{J}$ は電流密度，$\rho$ は電荷密度，$\mu_0$ は真空中の透磁率，$\varepsilon_0$ は真空中の誘電率，$c$ は光速を示す．Vlasov 方程式(1)を速度空間で積分すると，以下の電荷保存則が得られる．

$$\frac{\partial\rho}{\partial t}+\nabla\cdot\vec{J}=0 \tag{3}$$

Maxwell 方程式(2.1)に含まれる電流密度 $\vec{J}$ はプラズマの運動によって生じ，これにより電磁場が変化する．

電流密度 $\vec{J}$ は Vlasov 方程式(1)の第二項にあたる実空間の流束 $\vec{v}f_s$ を速度空間で積分することによって求まり，電流密度 $\vec{J}$ が電荷保存則(3)を満足する限り，Poisson 方程式(2.3)は自動的に満たされる．

以上の方程式は，Vlasov コードにおいて解いているプラズマ粒子の運動論方程式であり，無衝突プラズマの第一原理と呼ぶ．

### 2.2 数値解放の概要

Vlasov 方程式は 4 次元以上の「超次元」を扱う方程式であり，そのままの形で多次元数値積分を行うのは非常に困難であるため，演算子分離（operator splitting）法が古くから用いられてきた[1]. 過去の研究では，各次元(x, y, z, $v_x$, $v_y$, $v_z$)それぞれを 1 次元移流方程式に分解する方法が採用されていたが，本研究では，以下のように実空間移流，速度空間移流，速度空間回転の 3 つの物理的な演算子に分離する手法を用いている[2].

$$\frac{\partial f_s}{\partial t}+\vec{v}\frac{\partial f_s}{\partial \vec{r}}=0 \tag{4.1}$$

$$\frac{\partial f_s}{\partial t}+\frac{q_s}{m_s}\vec{E}\frac{\partial f_s}{\partial \vec{v}}=0 \tag{4.2}$$

$$\frac{\partial f_s}{\partial t}+\frac{q_s}{m_s}(\vec{v}\times\vec{B})\frac{\partial f_s}{\partial \vec{v}}=0 \tag{4.3}$$

この演算子分離は，PIC 法において Newton－Coulomb－Lorentz 式（荷電粒子の運動方程式）を時間 2 次精度で解く手法として広く用いられている leap-frog アルゴリズムに基づいている．

本研究では，演算子分離による数値拡散を抑制するために，多次元の線形移流方程式に対する演算子非分離（unspliting）法を新たに開発している[2]. また本研究では，無振動性及び正値性を保証するリミッタを新たに開発し，数値振動の抑制を行っている[3,4]．ここで無振動スキームとは，ある区間において新たな極値（極大，極小）を生じず，既に存在する極値は（できるだけ）減衰させないスキ

ームであり，ENO/WENO 法はこれに該当するが，TVD 法は極地を鈍らせるために該当しない．

式(4.3)は荷電粒子の速度が磁力線により運動エネルギーを保ったまま変化する回転方程式を表す．直交座標系における回転方程式は剛体回転問題と等価であり，線形移流問題と同様に，数値計算において最も基本的であるが，計算精度が重要となる問題である．本研究で採用している back-substitution 法[5]では，Boris アルゴリズム[6]に基づいて速度空間での粒子の軌道をバックトレースし，$v_x$, $v_y$, $v_z$ 方向それぞれの演算子を分離して回転運動を解いている．剛体回転問題では，系の外側，即ち速度空間において速度が速くなればなるほど移動量（加速）は大きくなり, Courant 条件の影響を受けやすくなる点に注意が必要であり, 今後, 陰解法や演算子非分離法の開発が必要である．

以上のように，Vlasov 方程式の数値解法は未だ発展途上である．この大きな原因は，Vlasov コードで扱う次元が多いためであり，開発やデバッグにために大容量の共有メモリ環境が必要となるからである．

一方，Maxwell 方程式(2.1)及び(2.2)は，FDTD（Finite Difference Time Domain）法と呼ばれる電磁場解析法を用いて解く．FDTD 法では，Yee 格子[7]と呼ばれる staggered 格子を用いており，式(2.4)が自動的に満たされるように物理量が配置されている．また leap-frog アルゴリズムに基づいて電場と磁場を半タイムステップずらしており，時空間精度は 2 次である．

### 2.3 ハイブリッド並列

Vlasov シミュレーションでは非常に多くのメモリを必要とするため，並列計算が必須となる．Vlasov コードで使用する物理量は全て格子点上で与えられており，並列化においては領域分割法が有効である．図 1 は実空間 2 次元及び速度空間 3 次元を使用する 5 次元 Vlasov コードにおける並列化の概念を示す．我々の目は 4 次元以上の空間を認識できないが，2 次元実空間の各格子上に 3 次元速度空間（速度分布関数）が定義されていると考えると分かりやすい．本研究では図 1 のように実空間（$x$−$y$ 平面）においてのみ領域分割を行い，速度空間の領域分割は行わない[8]．これは，電荷密度や電流密度などのモーメント量を計算する際に必要な速度空間の積分において，各実空間での reduction 処理を行わないようにするためである．

本研究グループの Vlasov コードでは，OpenMP によるスレッド並列も併用している．経験的に，Fujitsu FX シリーズにおいては，ハイブリッド並列のほうが flat-MPI 並列よりも効率的になる場合が多い．近年の Xeon プロセッサ（SandyBridge, IvyBridge など）においても，ハイブリッド並列のほうが flat-MPI 並列よりも効率的になるケースが出てきた．また，京コンピュータ 6144 ノードの実利用経験よ

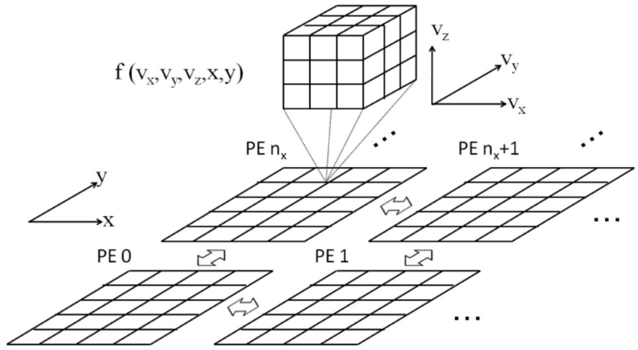


図 1 5 次元 Vlasov コードにおける空間領域分割[8].
Figure 1 The domain decomposition in the configuration space for the five-dimensional Vlasov code [8].

り, IO 処理や分散ファイルのデータ解析などの観点からプロセス数をできるだけ減らしたほうが利点は大きい．スレッド並列はそのオーバーヘッドの大きさから，できるだけより外側のループで行うのが効率的である．しかし, Vlasov モデルは 4 次元以上の超次元を扱い，メモリ使用量が非常に多いため，速度空間の格子点を $30^3$－$60^3$ に固定してコアあたりのメモリ使用量 1-4GB に設定しつつ，使用ノード数を増やして計算領域（実空間の格子数）を拡張していくのが実際の超並列計算機の利用方法である．近年の計算機においては，ノード内の共有メモリの容量は増えずにコア数のみが増加していく傾向にあるため，単一のループのみをスレッド化する単純な方法には限界がある．本研究グループの Vlasov コードでは，OMP DO ディレクティブの COLLAPSE オプションを最外側ループに挿入することにより，多重ループのスレッド化を行う．これにより，スレッド数を増やしたときに発生するオーバーヘッドを軽減することができる[9]．本研究で使用する 5 次元コードでは，$x$ 軸及び $y$ 軸の 2 次元についてスレッド並列を行う．

## 3. 計算機環境

本研究で使用した計算機環境は以下のとおりである．CPU として Xeon Phi 7250 (Knights Landing)を 1 つ搭載し，DDR4 の共有メモリを 96 GB 有する．この Xeon Phi プロセッサは 16 GB の Multi-Channel Dynamic Random Access Memory (MCDRAM)と呼ばれる高速メモリを有する．コア数は 68 であり, Hyper threading (HT)機能により 272 スレッドを同時実行できる．またコンパイラは Intel Parallel Studio XE Cluster Edition Ver.17.0.1.132 を搭載し，コンパイラオプションとして“`-ipo -ip -O3 -xMIC-AVX512`”を使用した．

MCDRAM は帯域 400 GB/s 以上を有し, DDR4 メインメモリ(~ 90 GB/s)よりも高速である. Xeon Phi KNL には，この

MCDRAM の利用方法として 3 つのメモリモードがある．Flat モードでは, MCDRAM と DDR4 メインメモリを 1 つの共有メモリとして使用する．Cache モードでは，MCDRAM を CPU の Level2 キャッシュと DDR4 メインメモリの間の Level3 キャッシュとして使用する．またこれらを混在した Hybrid モードも存在するが，本研究では利用しない．

また CPU コア群の分割方法として 5 つのクラスタモードがある．All2All モードでは，データがメモリアドレスに従って MCDRAM 上の一様に配置される．Hemisphere モードでは, CPU コア群を仮想的に 2 つに分割し，各 CPU コア群がアクセスするデータがより近い MCRDAM 上に配置される．Quadrant モードでは, CPU コア群を仮想的 4 つに分割される．SNC (Sub-NUMA Clustering)-2 モードでは，CPU コア群を 2 つに分割し, 2 ソケット CPU 構成として機能する．また SNC-4 モードでは CPU コア群を 4 つに分割し, 4 ソケット CPU 構成として機能する．

本研究で使用したメモリモードとクラスタモードの組合せを表 1 に示す．

表 1　本性能測定に使用したメモリモードとクラスタモードの組合せ

Table 1　Configurations of the Xeon Phi processor for the present performance measurement.

| 環境 | メモリモード | クラスタモード |
|---|---|---|
| #1 | Flat | All2All |
| #2 | Cache | All2All |
| #3 | Cache | Hemisphere |
| #4 | Cache | Quadrant |
| #5 | Cache | SNC-2 |
| #6 | Cache | SNC-4 |

## 4. 性能測定

まず，HT を未使用で 64 コアを使用した場合に，ノードあたりのスレッド数（プロセス数）を変えたときの経過時間(elapsed time)を計測した．つまり，本計測では MPI プロセス数とスレッド数の積は 64 で固定ある．計測に使用した格子数は Nx * Ny * Nvx * Nvy * Nvz = 128*64*40*40*40 であり，このメモリ使用量は作業配列を含めて約 28 GB であり，MCDRAM の容量を超える．この条件下で測定した 5 時間ステップの経過時間を図 2 に示す．

パネル(a)では Flat と Cache のメモリモードを比較しており, Cache モードを用いたときのほうが高速である（経過時間が短い）ことが分かる．また，Flat-MPI（64 プロセス・1 スレッド）の場合が最も性能が悪かった．

パネル(b)では All2All，Hemisphere，Quadrant 間のクラスタモードを比較しており，性能特性に大きな違いはないが，Quadrant モードが若干高速であった．

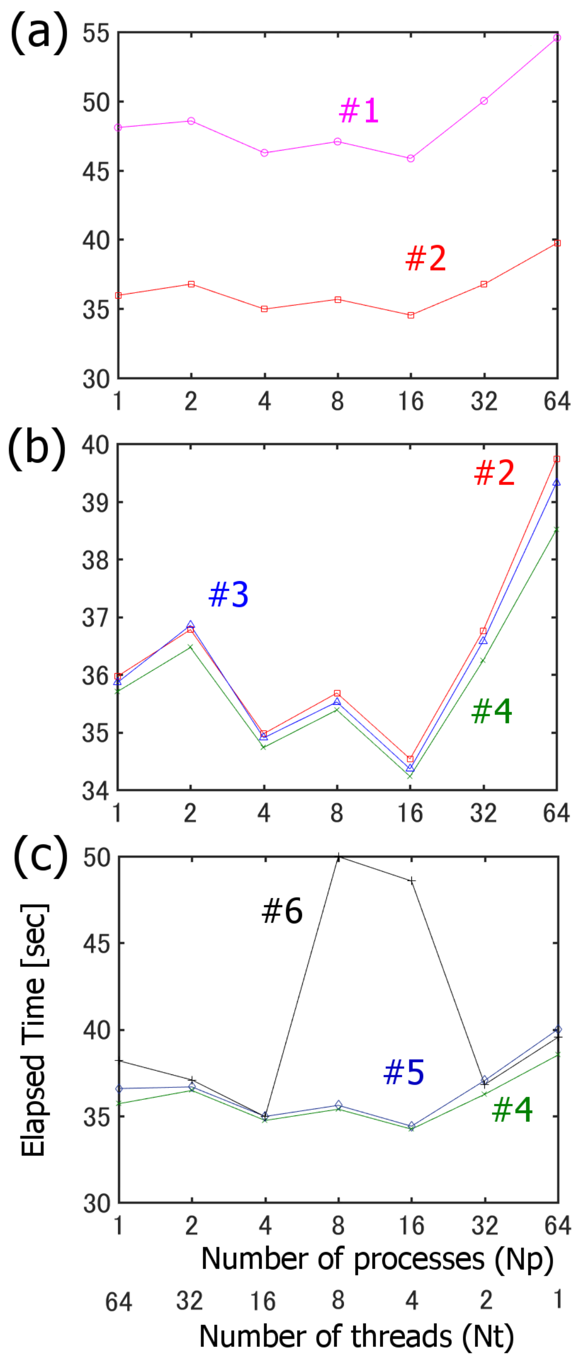


図 2　Xeon Phi KNL 1 プロセッサにおける Vlasov コードの性能特性. (a) Flat と Cache のメモリモード比較. (b) All2All, Hemisphere, Quadrant 間のクラスタモード比較. (c) SNC-2 と SNC-4 のクラスタモード比較.

Figure 2　Performance characteristics of the Vlasov code on a single processor of the Xeon Phi KNL. (a) Comparison between Flat and Cache memory modes. (b) Comparison among All2All, Hemisphere, and Quadrant cluster modes. (c) Comparison between SNC-2 and SNC-4 cluster modes.

パネル(c)では SNC-2 と SNC-4 のクラスタモードを比較しており（参考データとして Quadrant モードの結果も重ねている），SNC-2 モードは性能特性に大きな違いはないが Quadrant モードよりも若干遅く，SNC-4 モードでは 8 プロセス及び 16 プロセスを用いた場合に極端に性能が劣化する現象が見られた．これは MPI_sendrecv の通信時間が大幅に増えたことが原因であった．

図 3 および図 4 にそれぞれ，4 プロセス及び 16 プロセスを用いた場合にスレッド数を変更した場合の強スケール特性を示す．図 3 では Flat－All2All の場合を除き，17 スレッドまできれいにスケールしていることが分かる．17 スレッドを超えると HT を用いているが，Flat メモリモードの場合には，HT は効果がないことが分かる．一方，Cache モードの場合には，SNC-2 を除いて HT によって性能が若干改善しているが，どの場合でも 64 スレッド用いた場合のほうが 68 スレッド用いた場合よりも高速であった．また SNC-2 クラスタモードの場合に 32 スレッド用いたときのみ極端な性能の劣化が見られた．

図 4 では Flat－All2All 及び Cache－SNC-4 の場合を除き，4 スレッドまできれいにスケールしていることが分かる．4 スレッドを超えると HT を用いているが，図 3 と同様に Flat メモリモードの場合には，HT は効果がないことが分かる．また SNC-2 クラスタモードの場合に 8 スレッド用いたときのみ極端な性能の劣化が見られた．これは，プロセス数 Np*スレッド数 Nt=128 のときに SNC-2 では性能劣化が起こっていることを意味する．SNC-4 モードの場合には 4 スレッドおよび 8 スレッドを用いたときのみ極端な性能の劣化が見られた．しかし，17 スレッドを用いたときには SNC-4 が最速であった．

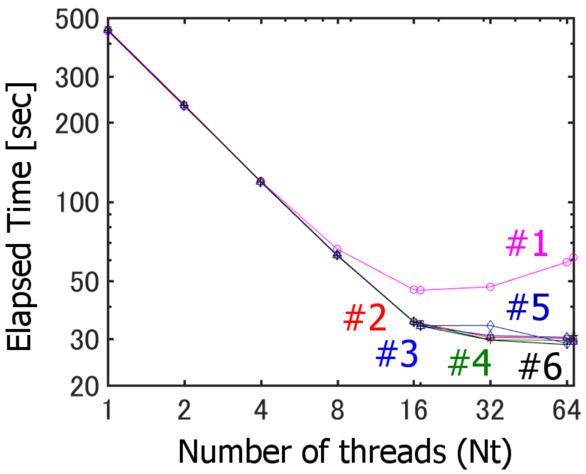


図 3　Xeon Phi KNL 1 プロセッサ・4 プロセスにおける Vlasov コードの強スケール特性.

Figure 3　Characteristics for the strong scaling of the Vlasov code on a single processor with 4 processes of the Xeon Phi KNL.

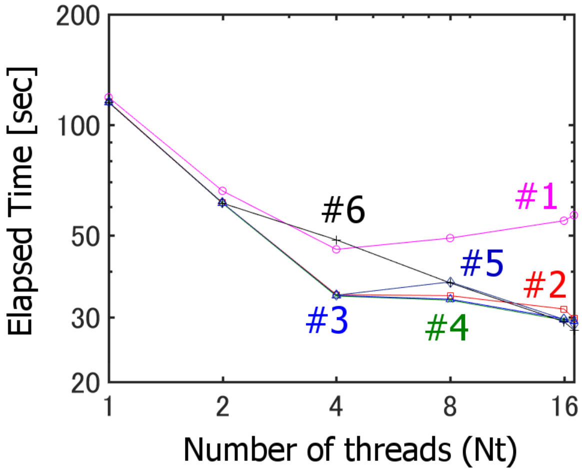


図 4　Xeon Phi KNL 1 プロセッサ・16 プロセスにおける Vlasov コードの強スケール特性.

Figure 4　Characteristics for the strong scaling of the Vlasov code on a single processor with 16 processes of the Xeon Phi KNL.

## 5. おわりに

Vlasov コードは，宇宙空間に広く存在する無衝突プラズマの第一原理シミュレーション手法である．プラズマは位置－速度位相空間における分布関数として定義され，超多次元のオイラー変数として与えられる．Vlasov シミュレーションは計算負荷が非常に高く，その手法の開発やデバッグが困難であるため，計算手法は未だ発展途上にある．本研究では，2 次元実空間及び 3 次元速度空間を扱う 5 次元 Vlasov コードについて，最新の CPU である Xeon Phi KNL (Knights Landing)において性能測定を行った．

まず, Vlasov コード Euler 型のコードでは，使用メモリ量が MCDRAM の容量を超えた場合に Flat メモリモードにおいて性能が劣化し，また HT も効果がないことが示された．Cache メモリモードを用いた場合には, All2All, Hemisphere, Quadrant のクラスタモード間で性能特性にほとんど差がないが，Quadrant が若干高速であることが分かった．一方，SNC-2 及び SNC-4 のクラスタモードを用いた場合に，特定のプロセス数とスレッド数の組み合わせで性能が極端に劣化する現象が見られ，コアに対するプロセス及びスレッドの配置とメモリアクセスが性能劣化に関係していることが示唆される．この性能劣化が環境変数で解決できるかどうかは今後の課題である．しかし, Xeon Phi 7250 プロセッサにおいて 272 の論理コアをフルに利用した場合，SNC-4 が最速であった．

**謝辞**　本研究は，科学研究費補助金 Nos.26287041 及び 15K13572 によりサポートを受けた．

## 参考文献

1. Cheng, C. Z., Knorr, G.: The integration of the Vlasov equation in configuration space, *J. Comput. Phys.*, Vol.22, No.3, 330—351 (1976).
2. Umeda, T., Togano, K., Ogino, T.: Two-dimensional full-electromagnetic Vlasov code with conservative scheme and its application to magnetic reconnection, *Comput. Phys. Commun*., Vol.180, No.3, 365—374 (2009).
3. Umeda, T.: A conservative and non-oscillatory scheme for Vlasov code simulations, *Earth Planets Space*, Vol.60, No.7, 773—779 (2008).
4. Umeda, T., Nariyuki, Y., Kariya, D.: A non-oscillatory and conservative semi-Lagrangian scheme with fourth-degree polynomial interpolation for solving the Vlasov equation, *Comput. Phys. Commun.*, Vol.183, No.5, 1094—1100 (2012).
5 . Schmitz, H., Grauer, R.: Comparison of time splitting and backsubstitution methods for integrating Vlasov's equation with magnetic fields, *Comput.Phys. Commun*., Vol.175, No.2, 86—92 (2006).
6. Boris, J. P.: Relativistic plasma simulation-optimization of a hybrid code, *Proc. Fourth Conf. Num. Sim. Plasmas*, ed. by J. P. Boris and R. A. Shanny, pp.3—67, Naval Research Laboratory, Washington D. C. (Nov. 1970).
7. Yee, K. S.: Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media, *IEEE Trans. Antenn. Propagat.*, Nol.AP-14, No.3, 302—307 (1966).
8. Umeda, T., Fukazawa, K., Nariyuki, Y., Ogino, T.: A scalable full electromagnetic Vlasov solver for cross-scale coupling in space plasma, *IEEE Trans. Plasma Sci.*, Vol.40, No.5, 1421—1428 (2012).
9 . Umeda, T., Fukazawa, K.: Hybrid parallelization of hyper-dimensional Vlasov code with OpenMP loop collapse directive, *Adv. Parallel Comput.*, Vol.27, 265—274 (2016).