

スーパーコンピュータ FX100 におけるブラソフコードの性能評価

梅田 隆行^{†1} 深沢 圭一郎^{†2}

ブラソフ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。ブラソフ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オイラー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 関連プロジェクトを通じて、ブラソフコードの性能チューニングを行ってきた。本講演では特に、スーパーコンピュータ Fujitsu FX100 システムにおける並列ブラソフコードの性能測定結果について報告し、これまでの FX シリーズにおける性能測定結果と比較する。

Performance evaluation of Vlasov code on the supercomputer FX100

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 supercomputer Fujitsu FX100. The performance comparison among the FX100 and other FX series is also made.

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ブラソフ（無衝突ボルツマン）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、ニュートンローレンツ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）を保つために特殊なデ

^{†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

一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つの手法であるブラソフ法は、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ブラソフ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、ブラソフ方程式は実空間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ブラソフ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における5次元ブラソフ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ブラソフコードの性能評価を行ってきた。本研究では、最新のスーパーコンピュータ Fujitsu FX100 におけるブラソフコードの性能測定を行う。また、Fujitsu FX1, FX10 及び京コンピュータ SPARC プロセッサを採用したFXシリーズ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、以下のブラソフ（無衝突ボルツマン）方程式によって記述される。

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

ここで $\vec{E}$, $\vec{B}$, $\vec{r}$ と $\vec{v}$ はそれぞれ電場、磁場、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マクスウェル方程式によって記述される。

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

$$\nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t} \quad (2.2)$$

$$\nabla \cdot \vec{E} = \frac{\rho}{\epsilon_0} \quad (2.3)$$

$$\nabla \cdot \vec{B} = 0 \quad (2.4)$$

ここで $\vec{J}$ は電流密度、 ρ は電荷密度、 μ_0 は真空中の透磁率、 ϵ_0 は真空中の誘電率、 c は光速を示す。ブラソフ方程式(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。

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

マクスウェル方程式(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 $\vec{J}$ はブラソフ方程式(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められ、電流密度 $\vec{J}$ が電荷保存則(3)を満足する限り、ポアソン方程式(2.3)は自動的に満たされる。

以上の方程式は、ブラソフ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

ブラソフ方程式は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]。

$$\frac{\partial f_s}{\partial t} + \vec{v} \cdot \frac{\partial f_s}{\partial \vec{r}} = 0 \quad (4.1)$$

$$\frac{\partial f_s}{\partial t} + \frac{q_s}{m_s} \vec{E} \cdot \frac{\partial f_s}{\partial \vec{v}} = 0 \quad (4.2)$$

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

この演算子分離は、PIC法においてニュートン-ローレンツ式（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（unsplitting）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、クーラン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、ブラソフ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、ブラソフコードで扱う次元が多いためであり、開発やデバッグの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、マックスウェル方程式(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

ブラソフ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。ブラソフ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 次元ブラソフ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点上に 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、各実空間でのリダクション処理を行わないようにするためである。

本研究グループのブラソフコードでは、OpenMP によるスレッド並列も併用している。経験的に、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ある。また、京コンピュータ 6144 ノードの実利用経験より、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

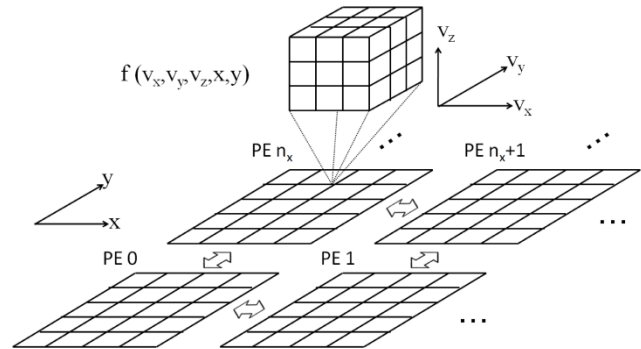


図 1 5 次元ブラソフコードにおける空間領域分割[8].

Figure 1 The domain decomposition in the configuration space for the five-dimensional Vlasov code [8].

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、ブラソフモデルは 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。本研究グループのブラソフコードでは、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]。

3. 性能測定

図 2 に、Fujitsu FX シリーズにおけるブラソフコードの弱スケリング性能測定の比較を示す。FX1, 京コンピュータ及び FX10 においては、コアあたりの格子点数を $20 \times 20 \times 40 \times 40 \times 40$ に固定した。これはコアあたり約 1GB のメモリ利用量に相当する。FX100 ではノードあたりのコア数に対するメモリ容量が減ったため、コアあたりの格子点数を $10 \times 20 \times 40 \times 40 \times 40$ に減らした。これはコアあたり約 0.5GB のメモリ利用量に相当する。また FX1 ではノードあたり 4 スレッド、京コンピュータではノードあたり 8 スレッド、FX10 及び FX100 ではノードあたり 16 スレッドを用いた。

図 2 はコア数に対する計算速度を示しており、性能がほぼスケールしていることが分かる。各システムにおける計算速度は、それぞれのシステムに搭載されているハードウェアカウンタを用いて計測した。本研究グループのこれま

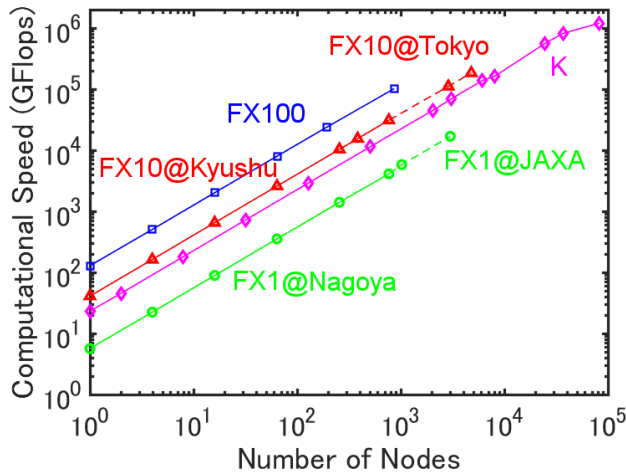


図 2 Fujitsu FX シリーズにおけるブラソフコードの弱スケール性能測定との比較. FX1, 京コンピュータ及び FX10 における計測結果は過去の研究から引用[9,10,11].

Figure 2 Computational speed of the Vlasov code on the Fujitsu FX series as a function of the number of compute node. The results on the FX1, the K computer and the FX100 systems were obtained by the previous studies [9,10,11].

での性能測定において, FX1 では約 13%の実効性能を得ており, 京コンピュータ及び FX10 では約 17%の実効性能を得ていた. またどのシステムにおいても 95%以上の並列効率(スケーラビリティ)を得ていた[9,10,11]. 本計測においては, FX100 での実効効率は約 11%に低下し, また並列効率は 864 ノード利用したときに約 90%であり, 実効性能と通信性能が FX10/京コンピュータと比べて大幅に低下した.

図 3 に, FX100 における性能測定の詳しい結果を示す. 上のパネルは, ノードあたりのスレッド数を変えたときの実効効率を示しており, 1 ノード(32 コア)から 864 ノード(27,684 コア)まで測定を行った. Flat-MPI(ノードあたり 1 スレッド・32 プロセス)の場合に比べて, スレッド並列を併用したハイブリッド並列のほうが性能が良いことが分かる. また, FX1, 京コンピュータ及び FX10 の場合と異なり, ノードあたりの最大スレッド(32 スレッド)を用いた場合は flat-MPI に次いで性能が低い. 1 ノードの場合はノードあたり 8 スレッドの場合が最も速く, ノード数を増やしていくとノードあたり 16 スレッドの場合の性能が逆転した.

下のパネルは, 1 ノード(32 コア)の計測結果に対する並列効率(スケーラビリティ)を示しており, 上のパネルと同じくノードあたりのスレッド数を変えて 1 ノード(32 コア)から 864 ノード(27,684 コア)まで測定を行った. 並列効率については, ノード数が少ないときに効率が悪かったノードあたり 32 スレッドの場合が, 864 ノード用いたときに最も並列効率が良くなった. ノードあたり 16 スレッドを用いた場合は概ね高い並列効率を示した. これらの結果は, プ

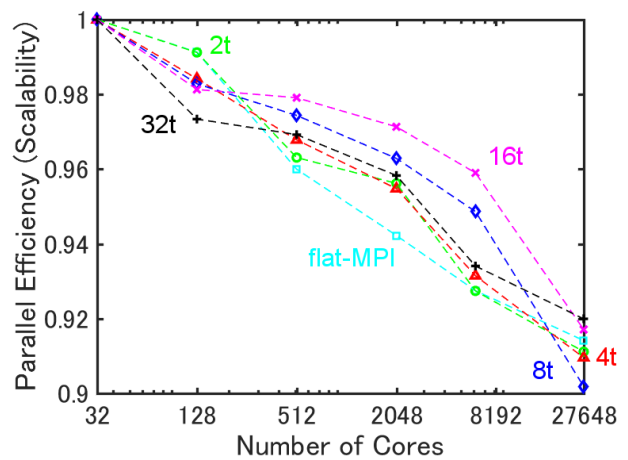
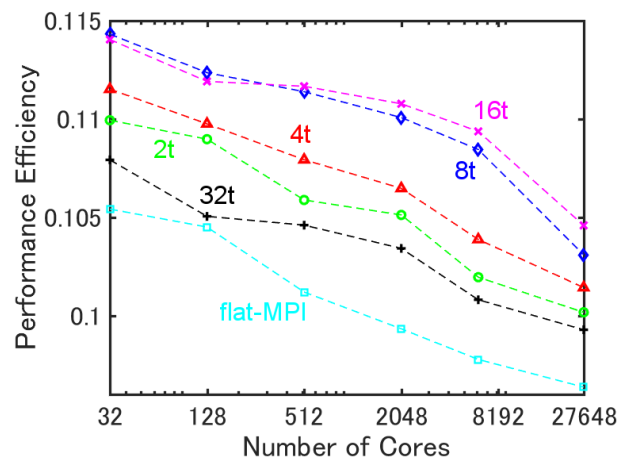


図 3 Fujitsu FX100 におけるブラソフコードの弱スケール性能測定. (上)ノードあたりのスレッド数を変化させたときの, コア数に対する実効効率. (下) ノードあたりのスレッド数を変化させたときの, コア数に対する並列効率(1 ノードの場合を 1 とする).

Figure 3 Performance measurement on the Fujitsu FX100. (top) The performance efficiency versus number of cores with different number of threads per compute node. (bottom) The parallel efficiency to the one-node case versus number of cores with different number of threads per compute node.

ロセス数が多くなった時に並列効率がより低下する傾向にあることを示唆している.

4. おわりに

ブラソフコードは, 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, 超次元のオイラー変数として与えられる. ブラソフ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次元ブラソフコードについて、最新のスーパーコンピュータである Fujitsu FX100 において性能測定を行った。これまでの FX10 及び京コンピュータに比べて実効効率が低下しており、ノード内演算のチューニングを行う必要があることが分かった。また通信性能に関してもこれまでの FX シリーズに比べてやや低下しており、原因について更なる調査が必要である。

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。ベンチマークテストに使用したスーパーコンピュータシステムの計算リソースは、HPCI システム利用研究(hp120092, hp140064, hp140081, hp150069 及び hp160015)、名古屋大学宇宙地球環境研究所 計算機利用共同研究、九州大学先端的計算科学研究プロジェクト及び、東京大学大規模 HPC チャレンジにより提供された。また性能チューニングに際し、サイエンティフィック・システムズ研究会ポストペタアプリ性能 WG に助言を頂いた。

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.*, No.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).
10. Umeda, T., Fukazawa, K.: Performance measurement of parallel Vlasov code for space plasma on scalar-type supercomputer systems with large number of cores, *Commun. Comput. Inf. Sci.*, Vol.402, 561—569 (2013).
11. Umeda, T., Fukazawa, K.: Performance tuning of Vlasov code for space plasma on the K computer, *Commun. Comput. Inf. Sci.*, Vol. 474, 127—138 (2014).