Vol.2010-HPC-124 No.8
2010/2/22

# 負荷分散技法 OhHelp による粒子・流体ハイブリッドプラズマシミュレーションの並列化

秋山 隼太[†1] 小路 真史[†2] 三宅 洋平[†3]
大村 善治[†2] 中島 浩[†3]

本論文では，粒子・流体ハイブリッドプラズマシミュレーションの，負荷分散技法 OhHelp を用いた並列化について述べる．すでに OhHelp を適用して良好な結果が得られている全粒子シミュレーションに比べ，ハイブリッドシミュレーションは電磁場の計算負荷が相対的に大きいため，計算と通信のバランスを大幅に見直した実装を行った．特に Cyclic Leapfrog 法による電磁場計算に関する，通信回数削減と計算量増加のトレードオフポイントを見出すことが可能な設計とした．また実用的なシミュレーションに不可欠な，スナップショットやダンプファイルの出力方式も検討し，並列 I/O を用いて並列性能と利便性の両立を図る設計・実装を行った．性能評価の結果，256 プロセスでの実行で 241-456 倍の台数効果が得られること，電磁場計算では計算量増加を抑えることが効果的であること，およびスナップショットと Weak/Strong Scalability との関係が明らかになった．

# A Parallelization of Particle-Fluid Hybrid Plasma Simulation with the OhHelp Load Balancer

Junta Akiyama,[†1] Masafumi Shoji,[†2] Yohei Miyake,[†3]
Yoshiharu Omura[†2] and Hiroshi Nakashima[†3]

This paper describes a parallel implementation of particle-fluid hybrid plasma simulation with our load balancing method OhHelp. In hybrid simulation, the cost to simulate the progress of electromagnetic field is more significant than that in full-particle simulation whose OhHelp'ed parallelization has already been proved efficient. Thus in this work we revisited the issue of the cost balance between computation and communication, especially for Cyclic Leapfrog method and the trade-off between reducing the number of communications and increasing computational amount. We also designed and implemented parallel-I/O for snapshot and dump, being essential for practical use of our simulator, to reconcile parallel performance and convenience of users. Our evaluation exhibited that the speedup with 256 process is 241- to 456-fold and that suppressing computational cost is the first priority in Cyclic Leapfrog. We also obtained valuable insights about the relationship between weak/strong scalability and snapshot frequency.

## 1. はじめに

プラズマシミュレーションは，宇宙のような広大な領域で運動するイオン・電子などの荷電粒子の振る舞いを調べるためには必要不可欠である．プラズマシミュレーションの方式は，主に対象とする現象の規模が大きい順から，イオンと電子の両方を近似的に流体として扱う MHD（Magneto-Hydro-Dynamics）シミュレーション，電子を流体近似しつつイオンを粒子として扱うハイブリッドシミュレーション，電子もイオンも粒子として扱う全粒子シミュレーションの３種類に分かれる．これらの中で後二者では電子やイオンを粒子として扱うため，数億 ～ 数百億個の莫大な数の粒子を処理しなければならない．そのためには巨大なメモリが必要となり，パーソナルコンピュータや小規模サーバはもちろん，1TB 級の大規模な共有メモリ型のスーパーコンピュータであっても，メモリ容量が不足するという事態が生じている．

したがって，大容量のメモリを比較的容易に利用可能な分散メモリシステムを用いたシミュレーションが不可欠であるが，分散したメモリに粒子をできるだけ均等に割り付けつつ，粒子と電磁場の相互作用を効率的に並列計算する負荷分散方式が要求される．そこで我々は，各プロセスが担当する粒子数と空間領域の大きさをスケーラブルに均衡化しつつ，かつ粒子と領域中の電磁場との相互作用を局所的に並列計算可能な負荷分散方式である OhHelp を提案している[1)]．OhHelp は領域を均等分割し，各々の部分領域とそれに含まれる粒子を各々のプロセスに割り当てる．この簡明な領域分割法では粒子の空間的な粗密による負荷不均衡が問題となるが，OhHelp では一つのプロセスを除く全てのプロセスが本来の担当とは別の部分領域を一つだけ担当し，その領域に含まれる粒子の一部分について電磁場との相互作用計算を行うことで，この問題を解決している．

†1 京都大学大学院情報学研究科
Graduate School of Informatics, Kyoto University
†2 京都大学生存圏研究所
Research Institute for Sustainable Humanosphere, Kyoto University
†3 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

我々はすでに OhHelp を全粒子シミュレーションに適用し，良好なスケーラビリティが得られることを確認している．一方，本論文で対象とするハイブリッドシミュレーションでは，電磁場の時間発展の計算負荷が相対的に大きく，粒子計算負荷の均衡を主眼としている OhHelp の有効性は明らかにされていなかった．一般に，プラズマ粒子は周りの電磁場によって軌道が変化し，その粒子の運動によって生じる電流により電磁場が変動するので，プラズマ粒子シミュレーションでは電磁場計算と粒子計算を交互に行うが，ハイブリッドシミュレーションでは粒子運動を計算するタイムステップごとに，電磁場の時間発展計算を複数タイムステップ行う必要がある．すなわち，流体近似する電子の速度はイオンの移動速度よりもはるかに大きいため，電子移動がもたらす電磁場の変化を高精度でシミュレートするには，電磁場計算のタイムステップを粒子計算のタイムステップよりも１桁程度細かくする必要がある．OhHelp のように領域分割を行う並列シミュレーションでは，電磁場計算のタイムステップを細かくすることは部分領域の境界値交換のための通信回数の増加を意味するため，複数のタイムステップを通信なしで計算できるように境界通信の対象である「袖」を大きくすることが考えられる．しかし，袖を大きくすることは通信量・計算量の増加も意味するため，通信回数削減とのトレードオフを考慮する必要がある．

本論文では，上記のように全粒子シミュレーションとは異なる特徴を持つハイブリッドシミュレーションに対する，OhHelp を用いた分散メモリ並列化について述べる．まず２章では，本論文の基礎となるハイブリッドシミュレーションを示す．続いて３章では，OhHelp を用いた分散メモリ並列化とそのために必要なプロセス間通信について，電磁場の時間発展計算のための境界値通信を含めて詳しく述べる．また実用的なシミュレーションでは不可欠となる，シミュレーション過程のスナップショットデータの出力についても述べる．４章では並列シミュレータの性能を，境界値通信のための袖領域の大きさの最適値や，スナップショットのためのファイル出力性能も含めて，詳しく述べる．

## 2. 粒子・流体ハイブリッドプラズマシミュレーション

### 2.1 基礎方程式

ハイブリッドシミュレーションは，イオンが主体となって起こるプラズマ物理現象について調べる場合に用いられ，イオンを粒子，電子を電気的中性を瞬時に保つための流体として扱うものである．このため，ハイブリッドシミュレーションでは電子の慣性項を無視し，光速よりも十分低速な運動に限定することで相対論的な効果も無視する．これらの前提に基づくハイブリッドシミュレーションの基礎方程式は，以下のものとなる．

$$\nabla \times \boldsymbol{E} = -\frac{\partial \boldsymbol{B}}{\partial t}, \tag{1}$$

$$\nabla \times \boldsymbol{B} = \mu_0 \boldsymbol{J}, \tag{2}$$

$$-en_e\boldsymbol{E} + \boldsymbol{J}_e \times \boldsymbol{B} - \nabla p_e = 0, \tag{3}$$

$$\frac{d\boldsymbol{v}_s}{dt} = \frac{q_s}{m_s}(\boldsymbol{E} + \boldsymbol{v}_s \times \boldsymbol{B}), \qquad \frac{d\boldsymbol{x}_s}{dt} = \boldsymbol{v}_s. \tag{4}$$

式 (1)，(2) は変位電流を無視した Maxwell の方程式である．$\boldsymbol{E}$ は電場，$\boldsymbol{B}$ は磁場，$\mu_0$ は真空中の透磁率，$\boldsymbol{J}$ は電流密度を表す．式 (3)，(4) はそれぞれローレンツ力に基づく電子，イオンの運動方程式である．ただし，電子を流体として扱うので，式 (3) は電子の慣性項を無視し，圧力項を考慮した運動方程式となっており，$e, n_e, \boldsymbol{J}_e, p_e$ はそれぞれ素電荷，電子密度，電子電流密度，電子流体の圧力を表す．宇宙におけるプラズマでは，プラズマ粒子が衝突する可能性は無視できるため圧力は断熱変化を仮定でき，

$$p_e = p_{e0}(n_e/n_{e0})^{\gamma} \tag{5}$$

で与えられる．ただし，$p_{e0}, n_{e0}$ はそれぞれ電子流体の圧力，電子密度の初期値であり，$\gamma$ は比熱比である．また式 (4) の $\boldsymbol{x}_s, \boldsymbol{v}_s, q_s, m_s$ はそれぞれ，イオン粒子の位置，速度，電荷，質量を表し，添え字の $s$ はイオンの粒子種を表す．

ハイブリッドシミュレーションでは，電荷は電子流体によって常に中性に保たれると仮定するので，以下の電荷中性条件が成立する．

$$-en_e + \sum_s q_s n_s = 0. \tag{6}$$

$n_s$ は $s$ 種イオンの密度である．また，この式 (6) と式 (2)，(3) より電場を求める式は以下のものとなる．

$$\boldsymbol{E} = -\frac{\boldsymbol{J}_i \times \boldsymbol{B}}{\rho_i} + \frac{(\nabla \times \boldsymbol{B}) \times \boldsymbol{B}}{\mu_0 \rho_i} - \frac{\nabla p_e}{\rho_i}. \tag{7}$$

以後，本論文中で $\boldsymbol{E} = \boldsymbol{E}(\rho_i, \boldsymbol{J}_i, \boldsymbol{B}, p_e)$ という記号が登場するが式 (7) の演算を意味することとする．ここで $\boldsymbol{J}_i = \boldsymbol{J} - \boldsymbol{J}_e$ はイオン電流密度，$\rho_i$ はイオン電荷密度を表す．イオン電流密度はイオン粒子の位置に基づいて，イオン電荷密度はイオン粒子の位置と速度に基づいて計算する．つまり，次式のように表される．

$$\boldsymbol{J}_i = \sum_s \sum_{\boldsymbol{x}_s \in R} q_s \boldsymbol{v}_s, \tag{8}$$

$$\rho_i = \sum_s \sum_{\boldsymbol{x}_s \in R} q_s. \tag{9}$$

```
for(t=0;t<timesteps;t++){
  ebfield();
  pressure();
  velocity();
  position();
  current1();
  charge();
  pressure();
  ebfield();
  current2();
}
```

**図 1** シミュレーションのメインループ
Fig. 1 Main Loop of Simulation

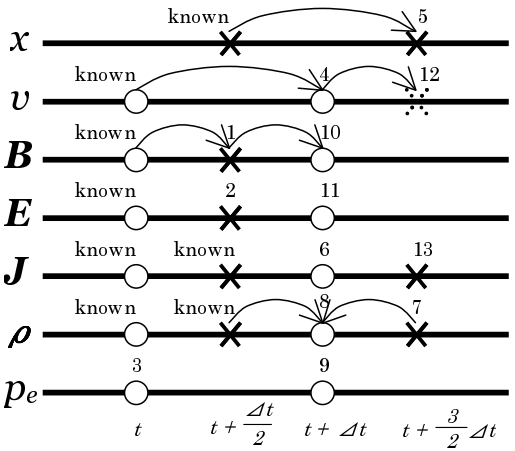


**図 2** メインループのタイムチャート
Fig. 2 Time Chart of Main Loop

### 2.2 計算の流れ

本論文での並列化の対象となるハイブリッドシミュレーションは，我々が逐次シミュレーションおよび共有メモリ並列シミュレーションのために開発したコード[5] に基づいている．このコードに用いられるアルゴリズムは，CAM-CL(Current Advance Method and Cyclic Leapfrog) 法[2] を基本としつつ速度計算に Buneman-Boris 法[3] を採用し，電流計算に関してはモーメントによる計算を排除することでエネルギー保存性を改良したものである．CAM-CL 法はハイブリッドコード向けのアルゴリズムであり，以下の２つの特徴を持つ．１つ目はマルチタイムステップを使用する，つまり軽い粒子，重い粒子，波動に対して異なるタイムステップを用いることができるということである．これにより，クーラン条件がもっとも厳しくなる波動の細かいタイムステップで粒子の運動を解き進める必要がなくなる．もう１つの特徴は，１タイムステップでの速度更新は一度ですむということである．ハイブリッドコードでよく用いられる predictor-corrector 法[3] では，１タイムステップで二度の速度更新を行なわなければならず，CAM-CL 法に比べて計算速度が劣る．

**図 1**，**図 2** にそれぞれ，このコードのメインループの概要とタイムチャートを示す．なお図 2 の数字は，各時刻での物理量を計算する順序を意味する．`ebfield()` は磁場 $\boldsymbol{B}$，電場 $\boldsymbol{E}$ をそれぞれ式 (1)，式 (7) に基づいて更新する．`pressure()` は電子流体の圧力 $p_e$ を式 (5) に基づいて更新する．`velocity()` と `position()` はイオン粒子の速度 $\boldsymbol{v}_s$ と位置 $\boldsymbol{x}_s$ を式 (4) に基づいて更新する．`current1()` および `current2()` はイオン電流密度 $\boldsymbol{J}_i$ を，`charge()` はイオン電荷密度 $\rho_i$ を，それぞれ式 (8)，(9) に基づいて計算する．

また，電磁場に対するクーラン条件によって制限されるタイムステップ幅は通常，粒子のそれよりも小さい．そのために，`ebfield()` で電磁場計算を行う際には，電磁場のタイムステップ $\Delta t_e$ を粒子のタイムステップ $\Delta t$ よりも細かく取って電磁場を解き進める Cyclic Leapfrog 法を用いる．Cyclic Leapfrog 法のアルゴリズムの概要は以下のようになる．

例えば，`ebfield` で電磁場を $t$ から $t + \Delta t/2$ 解き進めるとき，電磁場のタイムステップ $\Delta t_e$ を粒子のタイムステップ $\Delta t$ よりも $2k$ 細かく取るつまり，$\Delta t_e = \Delta t/(2k)$ として電磁場を解き進めるとする．まず空間方向に差分した式 (1)，(7) から $t + \Delta t_e$ の磁場 $\boldsymbol{B}$ と電場 $\boldsymbol{E}$ を次式で求める．

$$\boldsymbol{B}^{t+\Delta t_e} = \boldsymbol{B}^t - \Delta t_e \nabla \times \boldsymbol{E}^t, \tag{10}$$

$$\boldsymbol{E}^{t+\Delta t_e} = \boldsymbol{E}(\rho_i^t, \boldsymbol{J}_i^t, \boldsymbol{B}^{t+\Delta t_e}, p_e^t). \tag{11}$$

次に，$t + p\Delta t_e$ $(1 \leq p < k)$ での磁場と電場を次式のように偶数，奇数タイムステップ毎に更新していく．

Vol.2010-HPC-124 No.8
2010/2/22

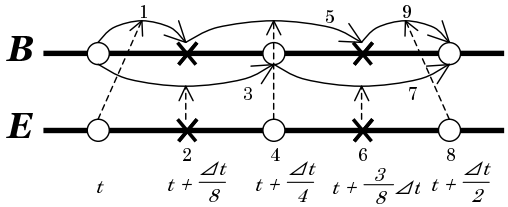


**図 3** Cyclic Leapfog 法のタイムチャート
Fig. 3 Time Chart of Cyclic Leapfrog

$$\boldsymbol{B}^{t+(p+1)\Delta t_e} = \boldsymbol{B}^{t+(p-1)\Delta t_e} - 2\Delta t_e \nabla \times \boldsymbol{E}^{t+p\Delta t_e}, \tag{12}$$

$$\boldsymbol{E}^{t+(p+1)\Delta t_e} = \boldsymbol{E}(\rho_i^t, \boldsymbol{J}_i^t, \boldsymbol{B}^{t+(p+1)\Delta t_e}, p_e^t). \tag{13}$$

そして，この式 (12) から求まった $t+k\Delta t_e$ での磁場 $\boldsymbol{B}^{t+k\Delta t_e}$ とは別に，次式で $t+(k-1)\Delta t_e$ での磁場から $t+k\Delta t_e$ での磁場 $\boldsymbol{B}_*^{t+k\Delta t_e}$ を求める．

$$\boldsymbol{B}_*^{t+k\Delta t_e} = \boldsymbol{B}^{t+(k-1)\Delta t_e} - \Delta t_e \nabla \times \boldsymbol{E}^{t+k\Delta t_e}. \tag{14}$$

最後に，上記で求めた偶数側と奇数側の磁場を足すことで，$t+\Delta t/2$ での値を得ることができる．

$$\boldsymbol{B}^{t+\Delta/2} = \frac{\boldsymbol{B}^{t+k\Delta t_e} + \boldsymbol{B}_*^{t+k\Delta t_e}}{2} = \frac{\boldsymbol{B}^{t+\Delta t/2} + \boldsymbol{B}_*^{t+\Delta t/2}}{2}. \tag{15}$$

上記のアルゴリズムの $\Delta t_e = \Delta t/8$ の場合のタイムチャートを**図 3** に示す．

### 2.3 空間格子の定義

空間微分を差分法によって近似するために，各物理量を空間格子へ配置する．空間格子は整数グリッドと半整数グリッドの 2 つを使用する．電場 $\boldsymbol{E}$ については空間格子の半整数グリッドに配置し，磁場 $\boldsymbol{B}$，イオン電流密度 $\boldsymbol{J}_i$，イオン電荷密度 $\rho_i$，電子流体の圧力 $p_e$ については整数グリッドに配置する．その様子を**図 4** に示す．図において黒丸は整数グリッド，白丸は半整数グリッドを表す．また，電磁場を計算するためには式 (1)，(7) のように電場 $\boldsymbol{E}$ と磁場 $\boldsymbol{B}$ の互いのローテーションが必要となるが，電場 $\boldsymbol{E}$ と磁場 $\boldsymbol{B}$ の定義位置を半整数ずらして格子に再配置することにより求めることができる．

一方，粒子の位置および速度に関しては，グリッドに配置せずに個々の超粒子が任意の値を持つことのできる PIC(Particle In Cell) 法を用いる．電荷，電流計算の際には，粒子の位置によって各グリッドへの電荷および電流の配分量を決める．

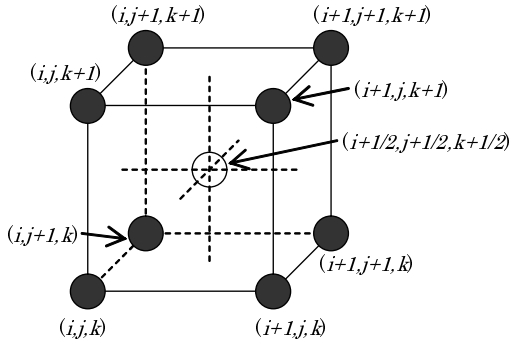


**図 4** 整数グリッドと半整数グリッド
Fig. 4 Integer and Half-Integer Grids

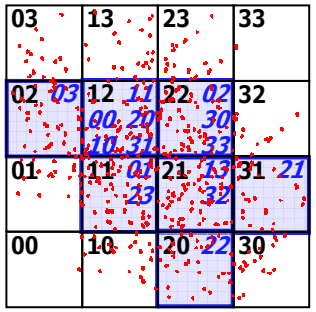


**図 5** OhHelp の領域分割
Fig. 5 Space Domain Partitioning in OhHelp

## 3. OhHelp を用いた並列化

### 3.1 OhHelp の概要

**図 5** に示すように，OhHelp は領域を単純に均等分割し，その分割した部分領域を各々のプロセスに「1 次担当領域」として割当てる．図で，黒色の数字はプロセス番号かつそのプロセスの 1 次担当領域の番号を示す．各プロセスの 1 次担当領域にある粒子の数が均衡していれば，すなわち領域 $s$ にある粒子の数 $P_s$ が全ての $s$ で次の不等式を満たすなら，それぞ

れのプロセスは1次担当領域とその中にある粒子を担当する．

$$P_s \leq (P/N)(1+\alpha) \equiv P_{limit} \quad (16)$$

ここで，$P$ は全粒子数つまり全領域にある粒子の合計，$N$ はプロセスの数，$\alpha$ は 0 より大きい数であり，$P_s$ の上限をプロセスごとの平均粒子数 $P/N$ よりも少し大きな値以下としている．このような粒子配置の状態でのシミュレーションを1次モードでのシミュレーションと呼ぶ．

一方，図 5 に示すように，不等式 (16) を満たさない領域が一つでも存在するときは，シミュレーションは2次モードで行われる．このモードでは一つのプロセス（図では 12 番）を除く全てのプロセスは，平均より多くの粒子が存在している領域を一つだけそれぞれの1次担当領域と共に担当する．例えば，図 5 において領域 22 は，青色で示されるプロセス 02，30，33 の「2次担当領域」であり，これらのプロセスは領域 22 に存在する粒子の計算を分担する．すなわち粒子数が平均よりも大きな1次担当領域を持つプロセスは，その領域を分担するプロセスに粒子の一部を「2次担当粒子」として委譲することで，自身が担当する粒子数を他のプロセスの担当粒子数と均衡させる．つまり，プロセス $n$ が担当する1次領域 $n$ の粒子数 $Q_n^n$ と，2次領域 $m$ の粒子数 $Q_n^m$ の合計 $Q_n$ が，式 (16) とよく似た次式を満たすように領域 $n$ を分担するプロセスに粒子を委譲する．

$$Q_n = Q_n^n + Q_n^m \leq (P/N)(1+\alpha) \equiv P_{limit} \quad (17)$$

ここで注意すべきは，一つのプロセス以外は2次担当領域を持つことであり，たとえば，プロセス 22 のように粒子数が平均以上の1次担当領域を持つプロセスであっても，別のプロセス（プロセス 20）の1次担当領域を2次担当することである．これは一見不合理であるが，担当粒子数を完全に均衡化するために必要である．また，稠密な1次担当領域を持つプロセスが他のプロセスの1次担当領域を分担しても不都合はまったくなく，2次モードの負荷は粒子の数だけでなく領域の大きさの点でも（1次モードの2倍であるが）均等となる．

負荷の均衡度の判定と不均衡である場合の均衡化は，図 1 の `position()` により粒子位置が更新されるたびに，以下のように行われる．

- 現在1次モードのとき
  - (1) 不等式 (16) が全ての領域で満たされれば，1次モードを継続する．このとき，隣り合う領域から移動してくる粒子を近接通信のみで移動する．
  - (2) 不等式 (16) が満たされなければ，2次モードへ移行し，全ての $n$ について $Q_n$ が完全に均衡つまり $P/N$ になるように2次担当領域を割当て，その割り当てにしたがって粒子を移送する．
- 現在2次モードのとき
  - (1) 不等式 (16) が全ての領域で満たされれば1次モードへ移行し，各領域中の粒子は領域を1次担当するプロセスに移送される．
  - (2) 不等式 (16) は満たさないが式 (17) を満たすときは，2次担当領域の割当を変更しない．この場合の粒子移送は，領域間の粒子移動と，領域を1次・2次担当するプロセス間の負荷均衡を保つためだけに行われる．
  - (3) (16) も (17) も満たさないときは，2次担当領域の割当を変更するが，なるべく割当を変えないようにする．

なお，2次担当領域の割当方法，不等式 (17) の充足判定法，ある領域を1次・2次担当するプロセス間で負荷均衡を保つための粒子移送などについては，文献 1) を参照されたい．

### 3.2 プロセス間通信

ハイブリッドシミュレーションに OhHelp を適用すると，以下のような計算でプロセス間通信が必要となる．

**電磁場計算** 3.1 節で述べたように粒子配置が不均衡なときは，各プロセスは他のプロセスの領域を2次担当領域として計算することとなる．2次担当領域では1次担当領域から委譲された粒子の運動や，その運動による電荷密度，電流密度を計算しなければならない．電荷密度，電流密度は粒子の位置と速度のみから計算できるが，式 (4) に示すように，粒子の速度を求めるためには電磁場が必要となる．しかし，2.2 節で示したように `ebfield()` での電磁場更新では，複数回電磁場を計算しなければならず計算コストが高い．さらに，電場更新には式 (7) に示すように電荷密度，電流密度や電子流体の圧力も必要となるので，2次領域で電場を計算するためには1次領域から2次領域へこれら3つの配列変数の放送が必要となる．そこでこれらを勘案し，電磁場計算は1次領域のみが行い，2次領域に粒子の移動を計算するために必要な最低限の電磁場を放送する方法を採用する．

この2次領域への放送には，OhHelp ライブラリ関数の oh3_bcast_field(prime, sec, fid) を利用することができる．ここで，第1引数は1次領域から放送する配列変数，第2引数は受け取る2次領域の配列変数，第3引数は放送を行う配列の識別子である．あらかじめ電磁場等の配列毎に通信の種類（境界，縮約，放送）と通信対象領域の位置とサイズを配列識別子と関連付けておくことで，メインループの中では単にこの関数を呼び出すだけで，電磁場の1次領域から2次領域への放送を行うことができる．

**電流・電荷密度計算** 電荷密度，電流密度の値は，2.3 節で示したように，粒子の位置によって各グリッドへ配分される割合が変わる．よって，1次領域と2次領域でそれぞれが担当

している粒子が影響を与える電荷・電流密度を独立に求め，２次領域で求めた値を１次領域に合計することによって，１次領域のみで担当する場合と同一の結果を得ることができる．

この合計には２次領域から１次領域への縮約通信である oh3_reduce_field(prime, sec, fid) が用いられる．ここで，第１引数は縮約される１次領域の配列変数，第２引数は縮約する２次領域の配列変数，第３引数は縮約対象となる配列の識別子である．

また，粒子の位置によって各グリッドへの配分量が変わるということは，各分割された部分領域の境界付近にある粒子によって，隣接領域のグリッドにも電荷密度，電流密度が配分されることになる．したがって，電荷・電流密度計算では，１次領域，２次領域で求めた値を１次領域に縮約したあとに，隣接プロセスに影響を及ぼす境界値付近の電荷密度，電流密度を隣接プロセスに加算するとともに，隣接プロセスが及ぼす影響を加算しなければならない．

そこで，OhHelp ライブラリの関数 oh3_exchange_borders(pri, sec, fid, ps) を用いて，隣接プロセス間で影響を及ぼす，また及ぼされた値を交換してそれぞれの領域に加算する．ここで，第１，第２引数はそれぞれ境界通信をする１次と２次領域の配列変数，第３引数は境界通信の対象となる変数の識別子，第４引数は２次領域でも境界通信を行うかを定めるフラグである．ここで，上記のように電荷・電流密度計算結果は１次領域に集約すればよいため，第４引数は偽，また第２引数は任意となる．

**粒子位置更新**　粒子の位置を更新するためには，その粒子の周辺の電磁場ベクトルがあれば十分である．また，各粒子間に直接の相互作用が存在しないため，粒子を１次領域と２次領域に分割しても独立に計算することができる．そこで，3.1 節で述べたように `position()` により粒子位置を更新するたびに oh3_transbound(ps,stats) を呼び出し，負荷の均衡度の判定とそれに基づくプロセス間の粒子移送を行う．ここで，第１引数は現在実行しているモード（１次または２次モード）を，第２引数は粒子統計処理の要否を，それぞれ定める．

### 3.3　電磁場計算のための通信

2.3 節で示したように，電磁場計算には電場・磁場相互のローテーションが必要となり，格子を半整数分ずらすことによりこれを解決している．このため，ある領域の磁場を更新するには座標値が小さくなる方向に，また電場を更新するには座標値が大きくなる方向に，それぞれ１グリッド分の余分な領域が必要となる．つまり一度の電磁場更新には，対象領域を囲む１グリッド分の境界領域が必要となる．OhHelp で領域分割された各部分領域でこの余分な領域を得るためには，隣接するプロセスとの境界値交換が必要となる．また，2.2 節で示したように，`ebfield()` では複数回電磁場計算が行われるため，細かいタイムステップで電磁場を計算する度に毎回隣接通信を行わなければならない．一般に通信のコストには，通信データ量に依存する部分と通信回数に依存する部分とがあり，後者の占める割合は必ずしも小さくない．したがって後者のコストを最小化するために,「袖」と呼ぶ複数グリッドからなる境界領域の通信を一度だけ行い，袖を含めた電磁場計算を袖領域をタイムステップごとに縮小しながら行う方法が考えられる．ただし，袖という余分な領域を計算するために，毎ステップ境界通信を行う場合よりも計算量が大きくなるというデメリットが生じる．さらに式 (7) より，袖付きの計算には圧力，電荷密度，電流密度も余分に必要となりこれらの袖を得るための通信も必要となる．今回の実装では，この通信量と計算量の最適なトレードオフポイントを与える袖の大きさを見極めるために，袖の大きさを１〜$k$ の範囲で任意に定めることができるように設計した．

### 3.4　ファイル出力

プラズマシミュレーションの重要な目的の一つは系の時間発展を調べることにあるため，あるタイムステップ毎の電磁場，電荷密度，電流密度等のスナップショットが必要となる．また，プラズマシミュレーションの実行時間は一般に長く，数週間や数ヶ月を要することも稀ではない．したがって実行中のトラブルによってシミュレーションが中止されても，それまでの実行が無駄にならないようにする必要があり，シミュレーション状態をたとえば１日に１回保存するための再開可能なファイルダンプが不可欠である．

今回の実装では，スナップショットやファイルダンプには HDF[6] ライブラリを用いた．バイナリ形式で出力されるファイルの内容は一般にプラットフォームやプログラミング言語に依存するが，HDF を使ったバイナリファイルはこれらに依存しないため，たとえば研究者の PC での解析のために要求されるデータの可搬性が保障される．また大量かつ複雑な構造のデータファイルを容易に生成でき，かつその一部分を取り出して解析することも容易であるため，時間方向に数多く生成されるスナップショットを１つのファイルにまとめて保存することができる．さらに，HDF5 からは MPI I/O による並列 I/O 機能がサポートされ，多数のプロセスに分割された電磁場などの領域データを，プロセス数に依存しない形で容易にかつ一定の並列効率で生成できるようになったことも大きなメリットである．

本論文のハイブリッドシミュレーションでは，プラズマの中で生起する波動現象を観測するので電磁場，電荷密度，電流密度のスナップショットが必要である．OhHelp により領域は分割されているが，各プロセスの１次領域の値を並列 I/O によって１つのファイルにま

とめて出力することで，プロセス数に依存しない形のスナップショットを得る．またダンプファイルには，再開時の初期値となる粒子，電磁場，電荷密度，電流，圧力の情報を保存する必要がある．ダンプファイルは解析には使用せず１つのファイルにまとめる手間は不要であるため，各プロセスがそれぞれの担当している一次領域の電磁場等を固有のファイルに出力して，それを再開するときは同一プロセスが読み込むようにする．ただし，粒子に関しては各ノードが担当している１次領域と２次領域の粒子を出力して，再開時にはそれらの粒子を全て１次領域の粒子として読み込む．この結果，粒子が不適切なプロセスに配置され，かつ全プロセスが１次領域の情報のみを持った状態で再開するが，再開直後に行われる負荷均衡処理により，適切な粒子配置とそれに基づく２次領域の設定が自動的に行われる．

上記に加え，系全体のエネルギーやプラズマ粒子の速度分布に関する情報を出力する．宇宙空間ではプラズマ粒子間のエネルギーの輸送は電磁場を介して行われるため，電磁場エネルギーおよび粒子の運動エネルギーの時間発展は，プラズマ不安定性など重要なプラズマ過程の進行状況を把握する上で有力な情報となる．また，粒子シミュレーションは，本質的に局所熱平衡状態から外れた（速度分布関数が Maxwell 分布から外れた）プラズマの振る舞いを再現できるところにその特長がある．こうした特長を最大限に活用して，現象の解析を行うために速度分布データの出力は必須である．

本研究では，エネルギーに関しては系全体の時間発展を観測することを目的とし，系全体の磁場エネルギーや粒子の運動エネルギーをあるタイムステップ毎に出力する．これは部分領域でそれぞれ計算したエネルギーを全プロセスで縮約することにより容易に実現できる．磁場エネルギーには磁場の情報のみが必要なので１次領域のみで計算し，また運動エネルギーは粒子が関係するため１次領域と２次領域でそれぞれ計算し，プロセス毎に１次と２次領域のエネルギーを加算した上で，全プロセスの値を縮約して出力する．なお，系全体のエネルギーを得るための加算順序は（少なくとも数学的には）任意であるため，粒子の運動エネルギーを１次・２次領域間で一旦縮約するような通信処理は不要である．一方，速度分布に関しては，分布情報のある平面上への射影など空間に依存する処理が必要である．そこで，エネルギー計算と同様にまず１次・２次領域ごとに計算した上で，OhHelp ライブラリの縮約関数を用いて１次領域に一旦縮約し，最終的に射影のための縮約を行った上で出力する．

## 4. 性能評価

### 4.1 計算環境と評価条件

OhHelp ライブラリを摘要した３次元粒子・流体ハイブリッドコードの性能評価を行った．計算には，京都大学の T2K オープンスーパーコンピュータである HX600 クラスタを使用した．HX600 は各ノードに４個の AMD 社製クアッドコア Opteron プロセッサとノードあたり 32GB（DDR2-667）のメモリを有する共有メモリマシンである．シミュレーションコードは Fortran90 で，また OhHelp ライブラリは C で記述し，それぞれ富士通のコンパイラ (ver. 3) を用いて最適化オプション -Kfast を指定してコンパイルした．

粒子配置に関しては，粒子を全空間に均等に分布させた場合と，$32\times32\times32$ の部分空間の中だけに分布させた不均等の場合（部分空間内の分布は均等）の２通りを計測した．粒子が $x$ 方向に一定の速度で進むように，電磁場，イオン電荷密度，電子流体の圧力を調整した．これにより，均等分布では１次モードのみの，また不均等分布では２次モードのみのシミュレーションが，それぞれ行われる．MPI プロセス数は $N=2^n (n=0\sim8)$ として各プロセスを各 CPU コアに割当て，領域は $p\times q\times r=N, (p:q:r)\in\{(1:1:1),(2:1:1),(2:2:1)\}$ となるように分割した．スケーラビリティの評価は，プロセス数に比例して全体の領域サイズと粒子数を増加させる Weak Scaling と，プロセス数に関わらず領域サイズと粒子サイズを一定に保つ Strong Scaling の二つの方法で行い，それぞれの問題パラメータは以下のように設定した．

まず Weak Scaling では，プロセスあたりの領域サイズを $32^3$ に，また格子点あたりの粒子数を 240 （プロセスあたりでは $15\times2^{19}$），タイムステップは 1600 とした．一方 Strong Scaling では，全空間を $64^3$ に固定し，格子点あたりの粒子数を 512（全粒子数は $2^{27}$），タイムステップは $N\leq4$ のとき $200N$，$N>4$ のとき 1600 とした．

また，式 (16)，(17) の各プロセスの粒子数が平均値をどの程度上回ることを許容するかの割合 $\alpha$ は 0.2 とした．

### 4.2 電磁場計算の性能

本節では，3.3 節で述べた電磁場計算のための通信による性能について述べる．今回実装したシミュレータでは粒子と電磁場の更新タイムステップの比率 $k$ を 8 としているので，境界通信の袖幅である $w$ を 1, 2, 4, 8 として，電磁場計算関数 `ebfield()` の 64 プロセスによる実行時間を測定した．ここで $w=1$ の場合は 1 サブタイムステップごとに 1 格子分の境界通信を行い，$w=8$ の場合は 8 サブタイムステップの計算の前に一度だけ 8 格子分の境界通信を行う．また，$w=1$ 以外のときは，電磁場以外にも電子流体の圧力 $p_e$，イオン電流密度 $J_i$，イオン電荷密度 $\rho_i$ の $w-1$ 個分の袖を確保する隣接通信が一度行われる．各プロセスが持つ領域 (=格子数) を $64^3$ として計算した結果を**図 6** に示す．

図 6 に示すように $w=1$ と $w=2$ がほぼ同等かつ最速であり，それ以上の $w$ では $w$ を

Vol.2010-HPC-124 No.8
2010/2/22

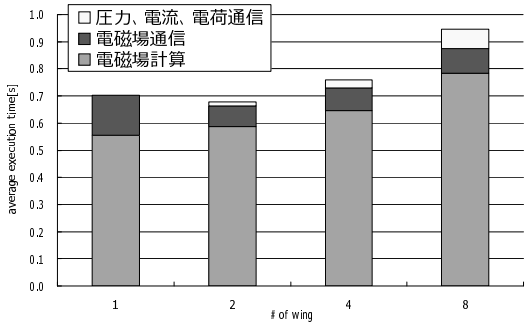


**図 6** 電磁場計算時間 (領域サイズ $= 64^3$)
Fig. 6 Execution Time to Update Electromagnetic Filed Values (subdomain size $= 64^3$)

大きくするに連れて計算時間が増加してしまうことが明らかになった．この理由の一つとして，たとえば格子数が $64^3$ の場合，$w$ あたりの通信データ量が約 200 KB と比較的大きく，1 回の通信に占める定数オーバヘッドが相対的に小さいことがあげられる．また我々のハイブリッドシミュレーションコードでは，$w > 1$ の場合に $w - 1$ に比例する圧力，電流・電荷密度の通信が必要となるため，大きな $w$ が一層不利になっている．この他，領域の一辺の大きさを $w$ 未満とすると通信が複雑化するなど，少なくとも我々のハイブリッドシミュレーションコードに関する限り，$w$ を大きくするメリットは乏しく，$w = 1$ がほぼ最適であると結論できる．

### 4.3 均衡・不均衡な粒子配置による Weak/Strong Scaling の性能

Weak/Strong Scaling の均衡・不均衡な粒子配置による性能を表 1，2 に，またこれらの値をグラフ化したものを**図 7** に示す．表では,「性能 / 逐次計算からの性能向上比」のように 2 つの事項を示している．性能の単位としては，以下に示される 1 秒間に移動計算を行った粒子数を用いている．

$$\frac{[\#\text{ of particles}] \times [\#\text{ of time steps}]}{[\text{exec. time excluding initialization}]}$$

Weak Scaling では，逐次性能は 0.88 Mparticle/s であり，表 1 と図 7 より，256 個のプロセッサを用いた場合，不均衡な粒子配置でも逐次性能と比べて 242 倍，均衡な粒子配置では 261 倍の性能を示し，空間領域サイズおよび粒子数の両者においてスケーラブルであることが確認できた．さらに，均衡，不均衡な場合も 256 プロセスまで良好な台数効果を

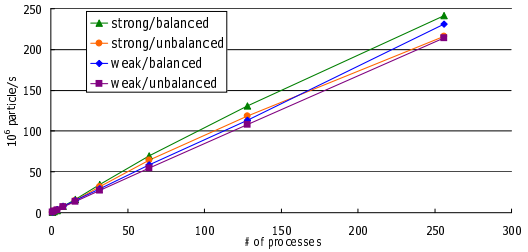


**図 7** シミュレーションの性能
Fig. 7 Performance of Simulations

**表 1** Weak Scaling の性能
Table 1 Performance of Weak Scaling Simulation

| #proc | balance | | unbalance | |
|---|---|---|---|---|
| 1 | 0.72/ | 0.82 | 0.72/ | 0.81 |
| 2 | 2.08/ | 2.35 | 1.96/ | 2.22 |
| 4 | 3.87/ | 4.38 | 3.55/ | 4.01 |
| 8 | 7.60/ | 8.60 | 7.02/ | 7.94 |
| 16 | 14.74/ | 16.66 | 13.75/ | 15.54 |
| 32 | 29.35/ | 33.18 | 27.40/ | 30.98 |
| 64 | 58.43/ | 66.07 | 54.40/ | 61.51 |
| 128 | 113.11/ | 127.89 | 107.93/ | 122.04 |
| 256 | 231.19/ | 261.39 | 214.00/ | 241.96 |

**表 2** Strong Scaling の性能
Table 2 Performance of Strong Scaling Simulation

| #proc | balance | | unbalance | |
|---|---|---|---|---|
| 1 | 0.37/ | 0.70 | 0.37/ | 0.70 |
| 2 | 1.15/ | 2.17 | 1.80/ | 3.40 |
| 4 | 2.81/ | 5.31 | 3.35/ | 6.32 |
| 8 | 7.66/ | 14.45 | 6.98/ | 13.17 |
| 16 | 16.42/ | 30.99 | 15.22/ | 28.72 |
| 32 | 34.52/ | 65.15 | 31.84/ | 60.10 |
| 64 | 69.62/ | 131.39 | 64.22/ | 121.20 |
| 128 | 130.97/ | 247.18 | 118.66/ | 223.93 |
| 256 | 241.66/ | 456.07 | 216.02/ | 407.68 |

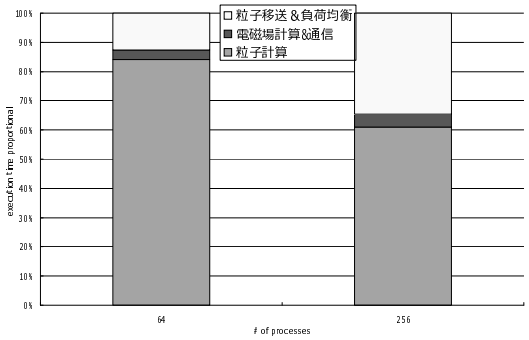


**図 8** Strong Scaling での計算時間の割合
Fig. 8 Exectution Time Breakdown in Strong Scaling Simulation

示している．

また，均衡と不均衡な粒子配置で性能が約 7% 下がった原因は以下の 3 つが考えられる．すなわち，平均よりも粒子数が 2 割まで多い場合を均衡として計算したことによる負荷の不均衡，1600 タイムステップに 180 回起こる 2 次担当領域の割り当ての変更による粒子の 1 次領域と 2 次領域の間の送受信，および 1 次領域から 2 次領域への電磁場の放送や 2 次領域の電荷・電流密度の 1 次領域への縮約のための通信である．

一方 Strong Scaling では，逐次性能は 0.53 Mparticle/s であり，表 2 と図 7 に示すように 256 プロセスの場合では，不均衡な粒子配置でも逐次性能と比べて 408 倍，均衡な粒子配置では 456 倍の性能を示した．また 2 プロセスから 64 プロセスまではほぼ線形の台数効果が得られているが，64 プロセスと 256 プロセスとの性能比は均衡な粒子配置で 3.5 倍，不均衡の場合では 3.4 倍とやや劣化している．ここで，この原因を示すデータとして均衡配置シミュレーションにおける，粒子の位置や速度を使用する粒子の位置更新や電荷・電流密度の計算にかかる粒子計算時間，電磁場の計算と境界通信にかかる時間，負荷が均衡しているか調べて粒子を（隣接）領域へ移送する時間の 3 種類の時間の割合を**図 8** に示す．

図 8 より，64 プロセスでは全体の 84% を粒子計算が占めており，粒子計算負荷の均衡に関して特に性能を発揮する OhHelp による並列化の恩恵が受けられることが分かる．また，256 プロセスでは粒子計算の割合は 61% まで減り，代わりに負荷均衡の調査と粒子移送にかかる時間の割合が 34% まで増えている．後者の時間のほとんどは粒子移送に費やされるが，一般にプロセスあたりの粒子数がプロセス数に反比例するのに対し，移送される粒子数はプロセス数の 2/3 乗に反比例するため，Strong Scaling ではプロセス数の増加に連れて粒子移送時間が相対的に増加するのは避けられない．なお今回の実験では粒子が特定の方向にのみ移動するため，64 プロセスと 256 プロセスでの移送粒子数の比が（$4^{2/3} : 1 \approx 2.5 : 1$ ではなく）2 : 1 であることも，粒子移送時間の比率増加の一因である．

Strong Scaling の均衡と不均衡な粒子配置で性能が下がった原因は Weak Scaling のときとほぼ同様と考えられる．ただし，16 プロセスから 64 プロセスのときは約 7.7% だった性能劣化が，128 プロセスと 256 プロセスのときそれぞれ，約 9.4%，10.6% と大きくなった理由は 2 次担当領域の割当変更回数の増加によるものと考えられる．すなわち，領域分割と粒子移動方向の関係から，16〜64 プロセスでは変更回数がそれぞれ 60 回，82 回，105 回であったのに対し，128 プロセスと 256 プロセスではそれぞれ 173 回，215 回に大きく増加し，割当変更に伴う全対全通信による粒子移送のオーバヘッドが顕著に現れているものと考えられる．

また，Strong Scaling と Weak Scaling の 256 プロセスでの絶対性能の差が均衡な場合で 4.3% に過ぎないにもかかわらず，逐次性能に対する比が 1.7 倍も異なる理由は，逐次計算と並列計算のワーキングセットサイズの比率が両者では大きく異なることである．Weak Scaling では逐次・並列計算でのワーキングセットサイズはほとんど変化しないが，Strong Scaling ではほぼプロセス数に反比例して減少する．この結果，たとえば逐次計算では粒子配列と CPU コアの親和性が保てないのに対し，2 プロセス以上ではコアが属するプロセッサに直結したメモリに確実に割り当てられる．また参照回数が粒子数に比例する電磁場および電流密度の配列が，逐次計算ではそれぞれ 12MB, 8MB であるため 2 次・3 次キャッシュに収容できないのに対し，16〜32 プロセスでは 2 次・3 次キャッシュに，64 プロセス以上では 2 次キャッシュのみで収容可能になることも，逐次・並列性能比が大きいことの要因である．なお粒子の存在範囲が限定されるとこれらの配列の参照局所性が高くなることが，Strong Scaling の 2 および 4 プロセスでは均衡配置よりも不均衡配置が高性能となる理由である．

### 4.4 HDF によるファイル出力性能

本節では，3.4 節で述べた HDF によるファイル出力のうち，電磁場データなどのスナップショットの性能について述べる．評価には，4.1 節で述べた均衡な粒子配置での Weak/Strong Scaling の設定を用いたが，Strong Scaling では 192 プロセス（$8 \times 4 \times 6$ プロセス）でのデータも測定するために全空間サイズを $64 \times 64 \times 48$ とした．またスナップショットの出力頻度は，Weak Scaling では毎ステップ，Strong Scaling では 1，2，5 および 10 ステッ

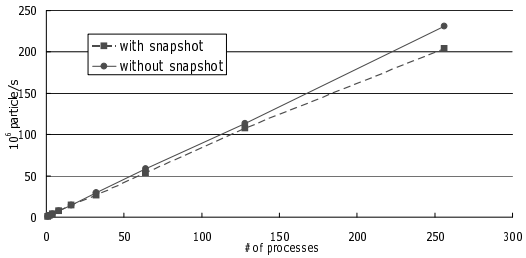


**図 9** ファイル出力と性能の関係 (Weak Scaling)
Fig. 9 Weak Scaling Simulation Performance with and without Snapshot

プごととした．

**図 9** と**図 10** はそれぞれ，Weak/Strong Scaling でのファイル出力の有無と性能の関係を示したものである．Weak Scaling では，プロセス数の増加によるファイル出力による性能劣化は小さく，スナップショットを毎ステップ出力しても実用上の問題はほとんどないことが明らかになった．実際のファイル出力時間は**図 11** に示すように，64 プロセス以上ではほぼプロセス数（すなわち出力データ量）に比例して増加するが，多数のプロセスからの出力を 1 ファイルにまとめるためのオーバヘッドは相対的に小さく，良好な性能が得られている．

一方，Strong Scaling では 1 タイムステップ毎や 2 タイムステップ毎にファイルを出力すると，192 プロセス程度で性能が飽和する．Strong Scaling では全体のファイルの大きさは等しく，プロセス数の増加に伴ってプロセス当たりの出力するデータ量が小さくなっていくので，全プロセスの出力を 1 ファイルにまとめるコストが顕在化しているものと考えられる．実際，ファイル出力時間は**図 12** に示すように，総データ量が一定であるにも関わらず 64 プロセス以上で急激に増加し，スケーラビリティの重大な阻害要因となることが明らかになった．

## 5. ま と め

本論文では，粒子・流体ハイブリッドシミュレーションに負荷分散技法 OhHelp を用いて並列化した．256 プロセスを用いたとき逐次実行と比べて 241-456 倍の性能を示し，電磁場の時間発展の計算負荷が相対的に大きいハイブリッドシミュレーションでも OhHelp が有用であることを明らかにした．また，Cyclic Leapfrog による電磁場計算の領域分割計算に

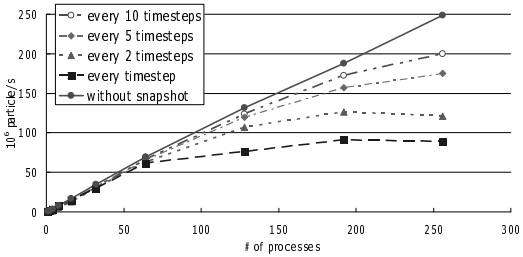


**図 10** ファイル出力と性能の関係 (Strong Scaling)
Fig. 10 Strong Scaling Simulation Performance with and without Snapshot

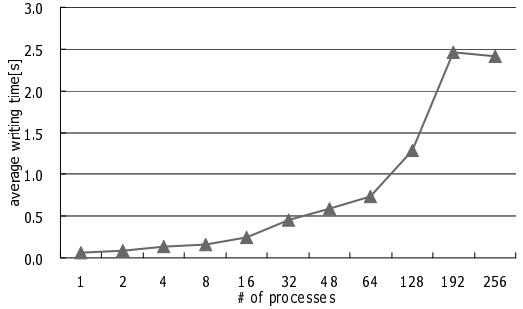


**図 11** ファイル出力時間 (Weak Scaling)
Fig. 11 Time to Output a Snapshot in Weak Scaling Simulation

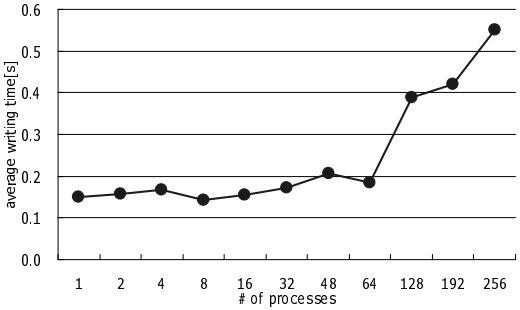


**図 12** ファイル出力時間 (Strong Scaling)
Fig. 12 Time to Output a Snapshot in Strong Scaling Simulation

ついて，境界通信の袖領域を大きくして通信回数を削減するよりも，袖を小さくして計算量の増加を抑える方が有利に働くことが明らかになった．さらに，電磁場，電荷密度，電流密度のスナップショットの性能に対する影響を調べ，Weak Scaling では大きな問題とはならないものの，Strong Scaling の性能に対しては重大な影響を及ぼすことも明らかにした．

このスナップショットによる性能劣化を抑止することは今後の重要な課題であるが，この問題に対するアプローチとして詳細に解析すべき現象のみに限定してスナップショットを取得する方法を検討している．一般に数万〜数十万ステップものシミュレーションにおいて，全ステップのスナップショットを解析することは非現実的であり，特に興味深い現象の周辺だけを詳細に解析できれば十分である．したがってこのような現象の兆候を捕らえることができれば，スナップショット間隔を適応的に制御することが可能となる．たとえばエネルギーの変化がしばしば兆候となることが知られており，時間的な変化量に応じたスナップショット間隔の調整が有力な方法として考えられる．また現象の発生後あるいはシミュレーションの終了後にしか解析対象を特定できないような場合には，ダンプファイルを利用した巻き戻しや所定時刻への移動を行った上で，スナップショット間隔を小さくした再実行を行うことも検討している．

**謝辞** 本研究の一部は文部科学省・科学研究費補助金#20300011 による．

## 参 考 文 献

1) H. Nakashima, Y. Miyake, H. Usui and Y. Omura: OhHelp: A Scalable Domain-Decomposing Dynamic Load Balancing for Particle-in-Cell Simulations. In *Proc. 23rd Intl.Conf.Supercomputing*, pp.90–99 (2009).
2) A.P.Matthews: Current Advance Method and Cyclic Leapfrog for 2D Multispecies Hybrid Plasma Simulations. *J.Comput.Phys.*, Vol.112, pp.102–116 (1994).
3) H. Matsumoto and Y. Omura: *Computer Space Plasma Physics*. Terra Scientific Publishing Company (1993).
4) H. Nakashima: OhHelp Library Package for Scalable Domain-Decomposed PIC Simulation, http://www.para.media.kyoto-u.ac.jp/ohhelp/ (2009).
5) M. Shoji, Y. Omura, B. T. Tsurutani, O. P. Verkhoglyadova and B. Lembege: Mirror Instability and L-mode Electromagnetic Ion Cyclotron Instability: Competition in the Earth's Magnetosheath, *J. Geophys. Res.*, 114, A10203, doi:10.1029/2008JA014038 (2008).
6) The HDF Group: Information, Support, and Software. http://www.hdfgroup.org/ (2009).