

平成 30 年 6 月 19 日現在

機関番号：13901

研究種目：挑戦的萌芽研究

研究期間：2014～2017

課題番号：26600152

研究課題名(和文)メニーコア演算器によるプラズマ粒子シミュレーションコード高速化への挑戦

研究課題名(英文)Challenge for code tuning of plasma particle-in-cell code using a manycore processor

研究代表者

齊藤 慎司(Saito, Shinji)

名古屋大学・理学研究科・特任准教授

研究者番号：60528165

交付決定額(研究期間全体)：(直接経費) 3,300,000円

研究成果の概要(和文)：本研究ではメニーコア演算器であるXeon Phiをプラズマ粒子シミュレーション(PIC)手法を用いた数値計算へ適用し、どの程度効率的な演算が可能であるのか検証を行った。PIC手法において最も計算コストが高いのが粒子の軌道および電流計算である。荷電粒子の運動計算については、スレッド並列およびベクトル化が実装可能なレベルに至り、効率的な演算が可能であることが検証出来た。しかしながら、電流計算については、空間に散らばる荷電粒子が不規則なメモリアクセスを行うためベクトル化の障壁となっている。空間的に不規則に散らばる粒子を動的にメモリ配置し、連続メモリアクセスを実現するのが今後の課題となる。

研究成果の概要(英文)：We study code tuning of plasma particle-in-cell (PIC) code for many core processor (Xeon Phi). In PIC method, calculations for charged particle motion and electric current density are the most expensive. It is the most important object for PIC method to reduce their calculation cost by using many core system. We have reduced the computational cost for particle motion, while there are some difficulty to reduce the cost for electric current density, because of random memory access of many charged particles for current density grids. It is the most important task for future study to redistribute particle data to realize sequential memory access that enable compilers to vectorize the calculation.

研究分野：プラズマ物理

キーワード：計算物理 宇宙物理 プラズマ粒子シミュレーション

## 1. 研究開始当初の背景

数値的にプラズマの非線形物理を記述するためにはいくつかの方程式系が存在するが、代表的なものとしては、プラズマを流体として扱う磁気流体方程式を解く MHD シミュレーションや、プラズマの位相空間密度の時間発展を解くブラゾフ方程式を扱う Vlasov シミュレーションなどがある。この中で本研究では、個々のプラズマ粒子の運動方程式を解くことによりプラズマ全体の運動を記述する粒子シミュレーション手法に注目する。このプラズマ粒子シミュレーション手法は、プラズマ粒子の運動論を含む計算方法であるが、非常に大量の粒子情報を保持し計算する必要があるため、大量のメモリおよび計算量を必要とする。このため、大規模な演算を行うためには多数の演算コアを用いた並列演算が必要不可欠となる。

近年 GPU や Xeon Phi といったメニーコア演算器が注目されてきている。これらはグラフィックカードと同じような位置付けで CPU 側と繋がっており、比較的 low 価格でありながら多数の計算コアを有している。これらを「アクセラレータ」として有効利用することで高効率演算が可能になると思われる。しかしながら、アクセラレータへの比較的低速なデータ転送が必要であり、搭載可能メモリ量が CPU 側と比べて劣るといったボトルネックが存在する。本申請者は平成 25 年度まで科研究費研究(挑戦的萌芽研究)において GPU を用いた粒子シミュレーション計算手法の開発に携わっており、「領域分割非同期転送手法」を確立し、このボトルネックの軽減に成功している。しかし GPU は CPU と異なる特徴を多く持っており、データ並列型の演算を非常に得意とするものの、メモリを共有するような演算(アトミック演算)をかなり不得意とし、本申請者は GPU では期待される粒子シミュレーション高効率化は難しいと判断した。しかし Xeon Phi は数世代前の CPU を利用していることもあり、CPU で行う場合と似たような計算手法で高効率化が望める。本研究では Xeon Phi を利用した粒子シミュレーションの高効率演算の実現に挑戦する。

## 2. 研究の目的

本研究では将来的に必要となりうるメニーコア演算手法について、プラズマ粒子計算手法への適用方法の開発およびその実装を主目的とする。特にこの計算手法で最も計算コストがかかる粒子の軌道および電流密度計算ルーチンに対して Xeon Phi 内でのスレッド並列およびベクトル化に着目して効率的な演算の実現を試みる。

また、メニーコア演算手法の開発で培ったベクトル化、スレッド並列、キャッシュチューニング手法等を用いて、既存の粒子シミュレーション手法の高速化を実現し、プラズマ粒子シミュレーションを用いた運動論的乱流に関する研究に活用し、学術的成果を得る。

## 3. 研究の方法

プラズマ粒子シミュレーション手法において、最も計算コストが必要となるのは、粒子軌道計算および電流計算である。それぞれの計算モジュールについて、キャッシュチューニング、ベクトル化、スレッド並列を順次適用していき、粒子データおよび粒子データから電磁場データへの効率的なメモリアクセス方法について検証を行っていく。

## 4. 研究成果

メニーコア演算器を用いた演算モデルは大きく以下 2 つに分類される。

(1) ホストからデータをすべてデバイス(メニーコア演算器)へ転送し、その中で計算された結果をホストへ返す。

(2) 常にホストとデバイスの間で通信を行い、平行してデバイス内で演算を行う

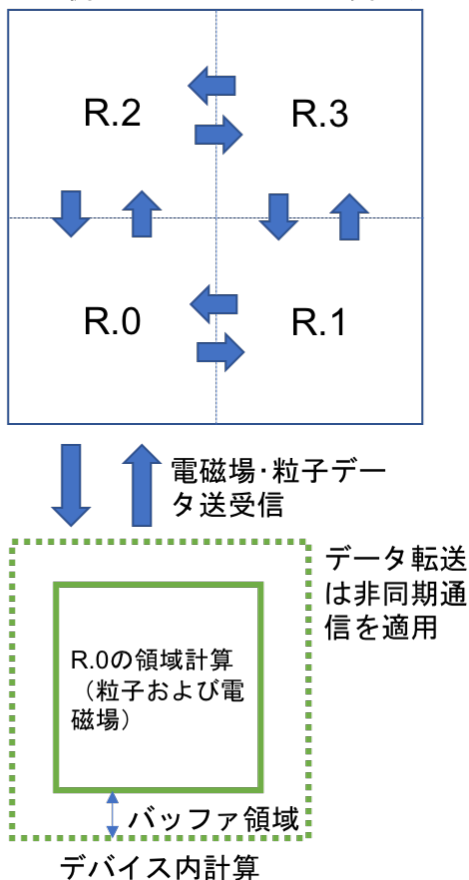
(1) は搭載出来るメモリに制限がある一方で、(2) はデータ転送速度と内部での計算速度に関連し、その必要量が変化する。(1) はデータ転送時間が演算時間より十分短ければ無視可能だがその転送時間は無くすることは出来無い。一方で、(2) の方法はデータ転送時間がデバイス内での計算時間より短ければ、データ転送とデバイス内での演算が平行に行われることで、データ転送時間が隠蔽されることになる。当初は(2)の手法がデータ転送時間の隠蔽という観点で重要であると考え、この手法で計算モデルを構築する予定であったが、実際転送時間と計算時間を比較したところ、単位計算量に対して(シミュレーション時間発展 1 ステップに相当させる)データ転送時間の方がデバイス内での計算時間より長くなる傾向があることがわかった。このため(2)の方法では転送時間が大きなボトルネックになり将来的な応用は見込めないと判断し、(1)の手法をベースとした方法について計算モデルの構築を行った。

(1) をベースとした計算モデルとして図 1 のようなモデルを考案した。まず計算領域全体を分割することで、領域分割法によるノード間並列を実現する。各ノードのメニーコア演算器が搭載されているとし、各領域(ここでは R.0 から R.3 を定義)が各々のデバイスと通信を行うことでメニーコア演算を実現する。図 1 では R.0 領域でのデータの流れを示す。R.0 空間領域に定義されたすべてのデータをデバイスへ転送し、デバイス内のデータの初期化を行う。ここで R.0 のデータはデバイス計算領域の緑の実線内の空間領域に相当する。緑の実線から点線域の間をバッファ域として定義し、初期の段階でこの領域にも電磁場を定義しておく。一方で粒子は初期の段階では緑の実線域内のみ配置される。これらのデータを用いて電磁場の発展方程式および粒子の運動方程式、また荷電粒子による電流計算が実行される。この際、バッファ域で変更された電流を含む電磁場情報は、ホ

スト・デバイス間通信および MPI によるノード間通信を介して更新を行う。一方で粒子データは毎ステップのデータ転送は実行せず、バッファ内の電磁場情報から時間発展を得る。この際、粒子からの場への影響は、電流場を介して他の領域へ伝えられる。バッファ内にいる粒子が外部境界へ達するステップ数以内に粒子データの転送処理を実行する。粒子転送処理時にバッファ内にある粒子情報をホストへ転送し、MPI 通信を介して隣接する空間領域へ転送する。これにより粒子データの転送コスト（ホスト・デバイス間の通信オーバーヘッド、バッファ内粒子の収集処理）を削減出来る。また、毎ステップ必要になる電磁場の転送処理は一部を非同期転送することがアルゴリズム的に可能であり、これにより転送時間の隠蔽を計る。これらにより効率的なホスト・デバイス間の通信を実現する。

メニーコアデバイス内ではスレッド並列およびベクトル化計算を効率良く実行することで高速化が期待される。電磁場の計算は基本的には前進差分と後進差分の組み合わせで実現されており、空間のインデックスを回す do ループに OpenMP 指示行を加えるのみで並列化が実現可能である。一方で粒子軌道に関係する演算は、1つの粒子に対する計算量が多いため、粒子インデックスの do ループをスレッド並列化するのみではなく、ループ内でのベクトル化が必須となる。Xeon Phi

ホスト側：MPIによるノード間並列



の場合、通常のプロセッサよりベクトル長が長く、適切に利用することで効率の良い演算が可能となる。ベクトル演算は粒子配列の配列演算として記述することでコンパイラが適用対象として判断することが可能で、適切な長さで配列演算を記述することで演算の効率化が期待出来る。しかしながら、各々の粒子位置での電磁場データが必要となるため、粒子位置がランダムに分布していると電磁場グリッドへランダムにアクセスすることになり、配列演算としての記述が出来なくなる。そのため演算内で同じグリッドにアクセスする保証がない場合は、ベクトル化されない do ループで記述する必要がある。この処理が一部ベクトル化の妨げとなるが、同様の計算をホスト側の1コアで計算するより、メニーコアデバイスにおいて10倍以上高速に演算が出来ることが確認出来た。

粒子演算は軌道計算のみならず、粒子の位置と速度の情報から各グリッドへ電流値を展開する処理が必要になる。この処理は粒子計算内でもっともコストがかかる演算で、粒子シミュレーション計算を効率化する上でのホットスポットとなる。しかしながら、粒子位置は電磁場の影響を受けて動くため、どの時間にどの位置にいるのかを予測することは困難であり、スレッド並列化により粒子演算を並列化すると、異なるスレッドが担う粒子が同じ電流場グリッドを同時に更新する可能性が生じる。通常このような演算は結果が予測出来なくなり、正しい解が得られない。これを解決する1つの方法としてはアトミック演算があるが、この処理は計算コストが大きいため非効率である。これを回避するために、演算器のメニーコア数の分だけ空間配列のレイヤーをもうけ、各コアが担って計算する電流場を独立に演算させる。これによりアトミック演算無しで、複数レイヤーに分けられた電流場が得られる。全粒子の演算を終えた段階で、各レイヤーの電流値をすべて統合し、全体の電流場を構成する。この際に空間グリッドのインデックスの do ループをスレッド並列で並列演算を行う。これにより電流計算のスレッド並列化は可能であるが、スレッド内で各電流レイヤーに電流値を与える場合、やはり粒子位置が連続で無いため、さまざまな電流場グリッドに不規則にアクセスすることになるため、ベクトル化およびキャッシュチューニングが効かず、結果効率的な演算が実現されない。ホスト側の演算と比較しても、それを大きく超える演算性能が得られなかった。

ここで最も重要な問題点は粒子位置が、任意の時間で各場（電場、磁場、電流場）のどのグリッドにいるのかが事前に決定出来ないことにある。直接的な解決方法の1つとしては、粒子の位置情報をもとに毎ステップに粒子配列全体にソーティングをかけ、連続アクセスを実現することにある。しかしながらこれはソーティング処理の計算コストが膨

大となり、現実的ではない。低コストにソーティングを実行するためには、ある特定のグループ内で連続アクセスを実現させるような部分的ソーティング方法が有効であると考えられる。粒子シミュレーションの場合は、各空間セル(隣接するグリッドに囲まれた領域)に含まれる粒子を1グループとし連続メモリ上に確保し、粒子が移動するごとに粒子の所属グループのメモリ領域に移動させる、ということで動的にソーティングを実現させる。類似した手法は2018年度JpGU国際会議の「宇宙プラズマ理論シミュレーション」セッションでの口頭発表でも粒子シミュレーションのメニーコア演算手法について開発を行っている研究者により発表されており、現状有効な手法であることが考えられる。

本研究では、メニーコア演算器を粒子シミュレーションに、どのように適用することが出来るか、またどの程度まで高速化が可能であるのかを検証するために取り組んできた。本研究を通して、スレッド並列化、ベクトル演算等の演算効率化手法について知見を得、これをもとにこれまで利用していた粒子シミュレーションコードの高速化および領域分割ノード並列の効率化への応用に至った。このコードを用いてプラズマの運動論的乱流の非線形発展に再現するプラズマ粒子シミュレーションを実施し、学会発表および論文発表に至った。

メニーコア演算器を外部デバイスとしてではなく、ノード内のホスト演算器として使用する計算機も増えてきている。このため、将来的にはメニーコアを意識した、スレッド並列およびベクトル化のコード開発技術がさらに重要になってくると思われる。前述した粒子セルごとの部分的なソーティング手法をベースにして、今後もメニーコアに対応する粒子シミュレーションコード開発を継続して実施していく予定である。

## 5. 主な発表論文等

〔雑誌論文〕(計 1件)

Saito, S., Y. Nariyuki, and T. Umeda (2017), "Generation of Intermittent Ion Acoustic Waves in Whistler Turbulence", *Phys. Plasmas* 24, 072304; doi: <http://dx.doi.org/10.1063/1.4990443> (査読有り)

〔学会発表〕(計 4件)

S. Saito, Y. Nariyuki, and T. Umeda, "Cascade of Whistler-mode Turbulence from Ion to Electron Kinetic Scales: Particle-In-Cell Simulation" (ポスター) AGU Fall Meeting 2017(2017年)New Orleans, USA.

S. Saito, Y. Nariyuki, and T. Umeda, "Generation of Intermittent Ion Acoustic Waves in Whistler Turbulence" (口頭) International Union of Radio Science General Assembly & Scientific Symposium, H35-2 (2017年) Montréal, Canada (Montréal's Convention Centre)

S. Saito, Y. Nariyuki, and T. Umeda, "Generation of Ion Acoustic Waves in Whistler Turbulence" (口頭発表) AGU Fall Meeting 2016 (2016年) San Francisco, USA.

齋藤慎司 "Xeon-Phiを用いたOFFLOAD版プラズマ粒子コード開発" (口頭) STE シミュレーション研究会(2015年)京都府京都市(京都大学)

## 6. 研究組織

(1)研究代表者

齋藤 慎司 (SAITO, Shinji)

名古屋大学・大学院理学研究科・特任准教授

研究者番号: 60528165