TSUBAME 共同利用　平成 26 年度　学術利用　成果報告書

利用課題名　TSUBAME2 GPU によるスピン系のクラスターアルゴリズム・モンテカルロシミュレーション
英文：Cluster algorithm Monte Carlo simulations of spin systems using TSUBAME2 GPU

利用課題責任者　岡部　豊
Yutaka Okabe

所属　首都大学東京理工学研究科
Graduate School of Science and Engineering, Tokyo Metropolitan University

物質内の相転移現象を解析するモデルとして，格子スピンモデルがある．格子スピンモデルの解析にはマルコフ連鎖モンテカルロ法がよく用いられており，我々はこれまでマルコフ連鎖モンテカルロ法の一つ，Swendsen-Wang　マルチクラスターアルゴリズムの GPU アルゴリズムの開発を行ってきた．本年度は今まで提案した方法の根本的な問題の解決に取り組み，シングル GPU およびマルチ GPU での新たなアルゴリズムを考案した．新たなシングル GPU のアルゴリズムでは従来のアルゴリズムよりも計算速度が 2 倍向上することを確認し，マルチ GPU のアルゴリズムでは 3 次元系の計算で問題であった GPU 間のデータ転送量を 90%削減した．

The classical spin model is one model for understanding mechanisms of phase transition. The classical spin model is often analyzed by using a Markov Chain Monte Carlo simulation, and we have developed the Swendsen-Wang multi-cluster algorithm, which is an algorithm of Markov Chain Monte Carlo simulations, with GPUs. In this paper, we identify the bottlenecks of previous methods with a single GPU and multiple GPUs, and we propose the new methods with a single GPU and multiple GPUs. As a result, the computation time of the new method with a single GPU is about half that of the previous method for all system sizes. In the new method of multiple GPU, the data communication between each GPU is reduced by 90%, and the number of communications between each GPU decreases by about half.

Keywords: Monte Carlo simulation, Union-Find algorithm, Ising model

背景と目的

物質内の相転移現象を解析するモデルとして，格子スピンモデルがある．格子スピンモデルを解析する標準的な手法として，モンテカルロ法は広く用いられており，局所的な情報を用いて更新が行われるメトロポリス法は格子スピンモデルに限らず，様々な分野で使用されている．メトロポリス法のようなシングルスピンフリップのモンテカルロ法には転移温度付近で緩和時間が急激に大きくなる問題がある．これを解決する一つの手法としてクラスターアルゴリズムが提唱されている[1]．クラスターアルゴリズムは Union-Find アルゴリズムを使用することで，スピンのクラスターを生成し，各クラスター内の全てのスピンを一度に更新するアルゴリズムである．そのため，局所的な情報ではなく，全体の繋がりの情報が必要となる．我々はこれまでクラスターアルゴリズムの 1 つ，Swendsen-Wang　マルチクラスターアルゴリズム[1]の GPU アルゴリズムの開発を行ってきた．本年度は今まで我々か提案してきた手法の根本的な改良を行い，シングル GPU[2]及び，マルチ GPU[3]の新たなアルゴリズムを考案した．

概要

シングル GPU の Union-Find アルゴリズムは大きく 2 つに分類される．一つは全体を小領域のブロックに分割し，各ブロックでのクラスターを形成後にブロック間のクラスターを接合するアルゴリズムである．もう一つは全領域のクラスターを直接形成する方法である．小領域でのクラスター形成と全領域でのクラスター形成のアルゴリズムとして，label equivalence アルゴリズム[4]が広く用いられている．このアルゴリズムは近接サイトとのラベルの比較と更新をラベルの更新がなくなるまで繰り返し実行するアルゴリズムである．しかし，この繰り

返しの際，更新しないラベルの計算も多数含まれており，この部分が無駄な計算となっていた．そこで，我々は更新する可能性があるラベルのみが計算を実行し，かつ近接サイトとのラベルの比較が一度で済む新たなアルゴリズムを考案した．この新たなアルゴリズムでは 4 つのカーネル関数を実行するのみで計算が完了する．このアルゴリズムの詳細については[2]の文献を参照して欲しい．

我々はマルチ GPU の Union-Find アルゴリズムを提案しており[5]，そのアルゴリズムを用いて格子スピンモデルの一つ，2 次元 XY モデルに対し，大規模計算を行った[6]．[6]の計算は 2 次元格子の計算であるため，GPU 間の通信の影響はそこまで大きくはなかった．しかし，3 次元の計算では通信の影響が大きくなり，GPU 間の通信が全体計算時間の 80%を占める[3]．そのため，高速な計算を実現するためにはこのデータ通信量を削減する必要があった．[5]で提案したアルゴリズムは各 GPU の袖領域ラベルを隣接 GPU に転送し，各 GPU 内のラベルを更新する方法を用いている．このデータ転送とラベルの更新は全ての GPU のラベル更新がなくなるまで繰り返される．この繰り返し回数は 50 を超え，袖領域のデータもこの回数分転送するため，データ通信の影響が大きくなっていた．そのため，我々は改良アルゴリズムとして，通信と更新の計算を行う前に各 GPU の袖領域部分の重複したラベルをまとめ，まとめたラベルのみを転送する方法を考案した．また，近隣領域を担当している GPU の重複したラベル情報を元に，各 GPU のラベルを再調整することで繰り返し回数の削減も行う．このアルゴリズムの詳細については[3]の文献を参照して欲しい．また，重複ラベルの抽出の際には逐次計算を使用する必要がある．逐次計算での抽出アルゴリズムでは袖領域の二乗の計算量が必要となるが，本研究では GPU 上で実装可能な重複ラベルの抽出アルゴリズムを新たに考案した[3]．このアルゴリズムは袖領域程度の計算量で抽出が完了する．

結果および考察

シングル GPU での結果を表 1，　表 2 に示す．表 1 は 2 次元イジングモデルの転移温度直上での 1 モンテカルロステップあたりの計算時間を示しており，表 2 は 3 次元イジングモデルの転移温度直上での 1 モンテカルロステップあたりの計算時間を示している．表 1，表 2 より，提案手法の計算時間は従来の label equivalence アルゴリズム[4]の計算時間の半分程度になっていることにわかる．

L	512	1024	2048	4096
従来手法[4]	2.99 ns	2.28 ns	2.09 ns	2.07 ns
提案手法[2]	1.29 ns	1.10 ns	1.01 ns	0.93 ns

表 1：2 次元イジングモデルの転移温度直上の計算時間

L	96	128	192	256
従来手法[4]	2.91 ns	2.35 ns	2.83 ns	2.38 ns
提案手法[2]	1.18 ns	1.13 ns	1.11 ns	1.12 ns

表 2：3 次元イジングモデルの転移温度直上の計算時間

[5]のマルチ GPU を用いた Union-Find アルゴリズムとマルチ GPU を用いた提案手法の比較を図 1 に示す．図 1 では 3 次元イジングモデルの転移温度直上の計算時間と繰り返し回数をプロットしており，1GPU 当たりの大きさは 512^3 である．図 1 の棒グラフは 1 モンテカルロステップあたりの計算時間を表しており，GPU 間クラスター形成の計算時間，重複ラベルの抽出とラベル再調整の計算時間，その他の計算時間毎に分割し表示している．また，図中の点は繰り返し回数を表している．図 1 より，GPU 間クラスター形成の計算時間が大幅に減少していることがわかる．また 1 反復当たりのデータ転送量は約 90%削減しており，反復回数も約半分になっている．

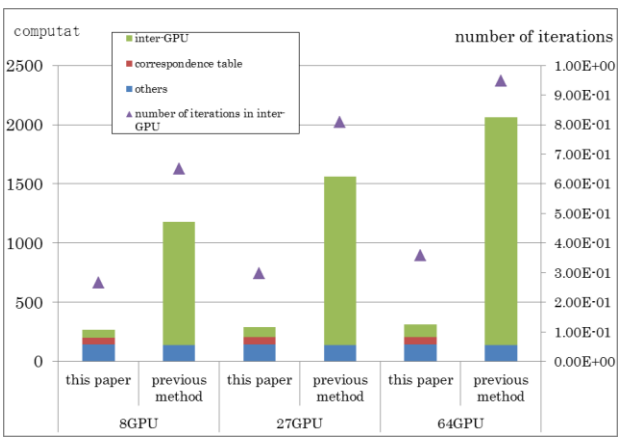

図 1：マルチ GPU を用いた 3 次元イジングモデルの転

移温度直上の計算時間と反復回数の比較

まとめ，今後の課題

本課題ではシングル GPU 及びマルチ GPU を用いた Union-Find アルゴリズムの改良に取り組んだ．従来提案していた手法の問題点を明らかにし，共有メモリ型のシングル GPU 及び，分散メモリ型のマルチ GPU での根本的な部分を修正した新たなアルゴリズムを提案した．

GPU を用いた Union-Find アルゴリズムの適用は正方格子，立方格子などの定形格子に限られている．しかし，本課題で提案したシングル GPU のアルゴリズムは任意格子への拡張が可能な手法であり，現在その拡張手法を考案している．この任意格子のアルゴリズムはマルチ CPU や Xeon Phi に適さない．これは任意格子のアルゴリズムはランダムメモリアクセスが主要な計算部分であるため，マルチコア CPU や Xeon Phi ではキャッシュミスやキャッシュコヒーレンスの影響によって計算時間が遅くなることが予想されるためである．

[1] R.H. Swendsen and J.S. Wang, Phys. Rev. Lett. 58 (1987) 86.
[2] Y. Komura, Comp. Phys. Comm. 194 (2015) 54.
[3] Y. Komura, (2015). Comp. Phys. Comm. 195, 84.
[4] O. Kalentev, A. Rai, S. Kemnitzb, and R. Schneider, (2011)　J. Parallel Distrib. Comput. 71 615.
[5] Y. Komura, and Y. Okabe, (2013). Comp. Phys. Comm. 184, 40.
[6] Y. Komura, and Y. Okabe, (2012). J. Phys. Soc. Jpn. 81, 113001.