

ループアルゴリズム量子モンテカルロ法の MPI/OpenMP による超並列化

松尾 春彦[†]

量子磁性体のシミュレーション方法の中で最も大規模なシステムサイズが扱えるのがループアルゴリズム量子モンテカルロ法である。この手法で計算の核となる部分は `union-find` アルゴリズムと呼ばれるグラフ理論のアルゴリズムであり、行列計算のような `flops` 値を要求する通常の科学計算と比較し、CPU に対しまったく異なる性能を要求する。本講演ではループアルゴリズムの実装を京コンピュータでのフルノード(64 万コア)での動作に向けチューニングしている現在進行中のプロジェクトについて紹介したい。

MPI/OpenMp Parallelization of Loop Algorithm Quantum Monte Carlo Method

HARUHIKO MATSUO[†]

The loop algorithm for quantum Monte Carlo simulations is a powerful tool to treat large quantum spin systems. The core of this algorithm is the union-find algorithm, which is popular in graph theory. But, compared to usual scientific calculations, such as matrix operation, the union-find algorithm requires of different type performance. In this talk, I will present our development of the loop algorithm, which is aiming to execute an extremely large-scale simulation on the 640,000 cores.

1. ループアルゴリズム量子モンテカルロ法

ループアルゴリズム量子モンテカルロ法[1]は量子磁性体のシミュレーション方法の中でも最も大規模なシステムサイズが扱える手法である。スピンを1つずつアップデートするような通常のモンテカルロ法のアルゴリズムでは物理的に興味を持たれる臨界現象のシミュレーションは難しい。それは通常の方法では臨界点付近での緩和時間が非常に長くなってしまい、サンプリングが上手くいかなくなり、正しい物理量を得られなくなってしまうからである。ループアルゴリズムではシミュレーションする系をいくつかのクラスターに分割し、それぞれ

をランダムにアップデートすることで緩和時間の短縮を行っている。そのため非常に大きな系の臨界点近傍のシミュレーションも可能となっている。

我々のグループではループアルゴリズムを大規模並列化し、これまで T2K 筑波や物性研の System B など約 10,000 コア規模の並列計算を行った。これらの成果として `spin-4` の Haldane ギャップおよび相関長が得られた。現在は `spin-5` の Haldane ギャップの評価に向け、京コンピュータ上でのプログラムの研究開発を行っている。

2. ALPS/looper プロジェクト

ループアルゴリズムを実装するにあたり

[†] 東京大学物性研究所
Institute for Solid State Physics, The University of Tokyo

ALPS[2]というオープンソフトウェアライブラリを利用している。ALPSは量子格子模型のシミュレーションを行うために格子構造や相互作用のXML形式の入力、計算結果のXMLやHDF5形式での出力、統計解析を行うための関数群、シミュレーションを行うためのスケジューラなどを提供している。また、量子格子模型に関する近年の代表的なシミュレーション法を実装したアプリケーション群も提供している。我々のグループが開発を行っているloopersはそのアプリケーションの1つである。

3. MPI/OpenMP による Union-Find アルゴリズムの並列化

ループアルゴリズムではスピンドウしをある規則により結びつけクラスターを生成し、最終的にはシステムをいくつかのクラスターに分割する。我々のループアルゴリズムの実装ではこの作業にunion-find アルゴリズムを利用している。Union-find アルゴリズムの計算量は $O(\alpha(n))$ 、 $\alpha(n)$ はアッカーマンの逆関数、となることが知られている。すなわちシステムサイズに対してほぼ線形な計算量となり、アルゴリズムの改良による計算量の改善はあまり望めない。そこで我々はシミュレーションで扱う時空間を分割する並列化により高速化を行った。虚時間方向ではMPIによる並列化を行っている。計算量は通信のコストを除けば虚時間の長さ按比例する。そのため計算量についてはMPIプロセス数で分割できる。メモリの使用量についても同様に分割される。一方、実空間方向についてはOpenMPによるスレッド並列化を行った。Compare and swap 命令を利用し非同期スレッド並列化[3]を行うことでこのアルゴリズムの高速化に成功した。1ソケットあたりではほぼコア数程度のスケーラビリティとなっている。

4. MPI 通信経路の最適化

ももとのMPI並列の通信経路は二分木状になるような実装を採用していた。京コンピュータのネットワークトポロジーは3次元トーラスであり、これに対応できるよう多次元方向への隣接通信が行えるアルゴリズムに変更した。

5. エクサスケールスパコンへ向けての課題

現在のスパコンはflops値至上主義と言える状況

です。Flops値の出ないアプリケーションの開発者からの観点からエクサスケールのスパコンに期待できることがあればお話ししたいと思います。

6. おわりに

本講演の内容はナノ分野グランドチャレンジ・大規模並列量子モンテカルロ法(開発責任者 藤堂眞治)で行ってきた研究開発に基づくものです。

参 考 文 献

- [1] H. Evertz “The loop algorithm” Adv. Phys., 52, 1, pp1-66, Nov 2010
- [2] B. Bauer et al. “The ALPS project release 2.0: open source software for strongly correlated systems” J. Stat. Mech., P05001 (2011)
- [3] 松尾春彦, 藤堂眞治, “Union-find アルゴリズムの非同期スレッド並列化”, HPCS2011 論文集