

## 課題名(タイトル):

## ポスト「京」時代に向けた格子 QCD アプリケーションの最適化と性能評価

利用者氏名:○中村宜文

理研における所属研究室名:計算科学研究センター フラッグシップ2020プロジェクト アプリケーション開発チーム

## 1. 本課題の研究の背景、目的、関係するプロジェクトとの関係

本研究は、ポスト「京」時代に利用可能な汎用スーパーコンピュータ上で格子 QCD アプリケーションが高効率に動作するため必要な要素技術と、最適化手法を明らかにすることを目的とする。格子 QCD アプリケーションは、ポスト「京」の重点アプリケーションに選ばれており、ポスト「京」時代においても、基礎物理学分野におけるその重要度は極めて高い。ポスト「京」時代に、現在の格子 QCD アプリケーションが高効率動作することは困難であると考えられており、これを克服するための要素技術と最適化手法の開発が求められている。また、高性能計算技術は日々進歩しており、最新のスーパーコンピュータなくして本研究を遂行することは不可能である。

## 2. 具体的な利用内容、計算方法

格子 QCD は 4 次元時空を規則格子に分割し、格子点と格子点を繋ぐ辺上に物理自由度において量子色力学 (QCD) の基礎方程式を評価する。格子 QCD で最も時間が費やされる部分はクォーク伝搬関数を解くところで、これはグルーオン背景場に依存するディラック方程式を 4 次元格子上で離散化した大規模連立方程式を解くことに相当する。これを QCD クォークソルバーと呼ぶ。離散化手法は幾つか存在するが最もよく使われるウィルソン型の離散化では大規模連立方程式の係数行列は規則格子に基づく大規模ステンシル疎行列となる。クォーク伝搬関数を解くための大規模連立方程式は反復法で解かれる。格子点上には未知数としてクォーク場に相当する 24 実数、格子辺上にはグルーオン場に相当する 18 実数が配置されており、係数行列は 1 階差分型のステンシルの構造を持ちメモリバンド幅律速となっている。メモリ容量としては比較的少ないものとなっている。

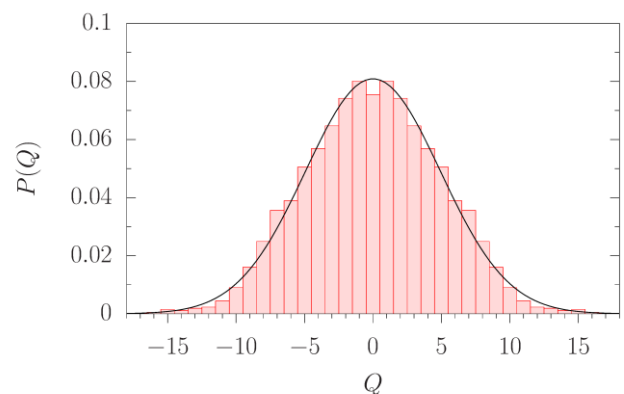
ポスト「京」時代には、512Bit の SIMD 幅が主流になり、メモリバンド幅は現在より狭くなると予想される。したがって、ポスト「京」時代のスーパーコンピュー

タに置いて格子 QCD アプリケーションは高効率に実行するためには、512bit の SIMD 幅を有効に使い、メモリアクセスを減らす工夫が必要であると考えられる。BW-MPC の CPU は skylake であり、AVX512 と呼ばれる 512bit 幅のベクター演算命令セットがある。このため、ポスト「京」時代のスーパーコンピュータ向けのアプリケーション開発に最適と言える。

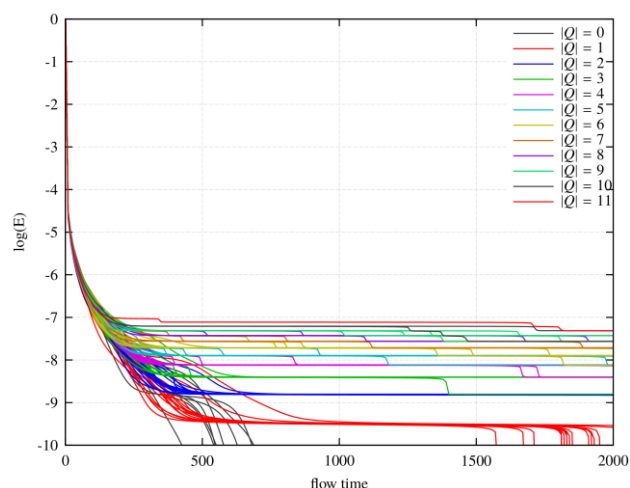
本研究課題では、主に BW-MPC において、格子 QCD アプリケーションの最適化と性能評価を行う。本年度は、グラディエントフローと呼ばれる一種の拡散方程式によってゲージ場を変形していく手法を用いて、トポロジカルチャージと呼ばれる位相不変量を計算し、トポロジカルチャージごとに結合定数を算出する手法の性能評価を行う。

## 3. 結果

テスト配位として  $24^4$ 、 $32^4$  格子をそれぞれ 5000 トラジェクトリ生成し、グラディエントフローを行った。下図は、 $24^4$  格子の逆結合定数  $\beta=6$  のトポロジカルチャージの分布である。Q=0 を中心にガウス分布しているのが分かる。



グラディエントフローを用いて、トポロジカルチャージを測定するには、拡散方程式に従って、ゲージ場の時間発展を  $\delta t=0.01$  で数百から数千ステップ行う必要がある。さらに下図のように、トポロジカルチャージの絶対値に比例する古典的なエネルギーを得るには、flow time が 1000、すなわち、十万ステップ分のフローが必要である。



この計算を、BW-MPC の 32 ノードで実行すると 1 配位あたり 8000 秒かかる。そのうち拡散方程式のステンシル計算に必要な袖領域の通信に 3000 秒程度かかる。

#### 4. まとめ

グラディエントフロー法の解析は、1 配位あたり、100 ノード時間積という大きな計算コストがかかる。グラディエントフロー計算の演算部分は複素数演算で、データ構造は array of structure のデータレイアウトを採用しているため、SIMD 化されていない。そのため、において、演算の SIMD 化と通信の隠ぺいが今後の課題である。

#### 5. 今後の計画・展望

グラディエントフロー計算に係るデータの配列を変更して演算部分の SIMD 化、演算と通信のオーバーラップによる通信オーバーヘッドを減らす最適化を行う予定である。