

受付 ID	16a42
分野	素粒子

格子 QCD 共通コードの超並列 GPU・メニーコアクラスタ計算機へ の実装

Implementation of Lattice QCD common code to large scale parallel supercomputer with GPU and manycore acceleration

根村英克

大阪大学核物理研究センター

1. 研究目的

格子 QCD 共通コードプロジェクトは、2008 年より発足した新学術領域研究「素核宇宙融合による計算科学に基づいた重層的物質構造の解明」の中の A04 班「分野横断的アルゴリズムと計算機シミュレーション」のプロジェクトのひとつとして開始された。そこで掲げられた目標は、『(a) 初心者にも使いやすく、理解可能であるシンプルな構造を持つこと。(b) 修正、追加が簡単に出来ること。(c) 検証や管理が容易であること。(d) 並列化や演算加速機などにも対応していること。(e) 十分な計算スピード(効率)を持つこと。特に、計算機固有の最適化も可能であること。』である。この格子 QCD 共通コードは、2012 年 7 月に最初の公開版が発表された。最新版は、2017 年 3 月に公開されたバージョン 1.4.0r2 である。

本共通コード開発プロジェクトの活動の大きな柱のひとつは、共同利用可能な大型計算機を具体的なターゲットとして、その性能を引き出せるようにコードの開発を継続し、格子 QCD 計算を軸とした様々な研究が円滑に進めていけるような基盤環境を整備していくことである。従って、本学際共同利用プロジェクトへの応募の目的は、(1) HA-PACS など演算加速機構を持つ大型計算機の性能を引き出すようなコードの開発を行うことにより、格子 QCD 共通コードの有用性を高めること、(2) それに伴って、格子 QCD 共通コードのユーザー数を増やすこと(格子 QCD 計算を主としている研究者だけでなく原子核や宇宙など関連分野の研究者が格子 QCD 計算を理解するための教材的な役割も含める)、(3) ひいては、原子核や宇宙などの関連分野を含めた日本における基礎科学研究体制の層の拡大・充実に資することである。

2. 研究成果の内容

2017 年 3 月 15 日に ver.1.4.0 を公開した。また、活動開始以来、国際会議等で講演発表を行い、会議報告書を発表している。2016 年度では、OFP 試験利用の成果を

中心として講演発表を行った。データレイアウト変更など OFP 向けコード最適化により、格子サイズ次第ではあるがアセンブラ版(P.Boyle 氏らによる Grid コード)と同程度の性能を出すためのノウハウが蓄積された。

なお、共通コードを使用した研究論文が、今年度新たに 11 本追加された。通算 24 本の論文が共通コードを元に発表されている。これらの情報は、以下のページ [<https://www.bridge-hpc.org/dokuwiki/>]から参照できる。

3. 学際共同利用として実施した意義

他分野で盛んに行われているように、格子 QCD 計算においても、GPU のような演算加速機構を備えた大規模並列計算機を活用する例が増えてきている。ただし、QUADA に代表されるように演算加速機構に専用のコードを使用するが多い。それに対して、我々がやっている格子 QCD 共通コード開発プロジェクトでは、スーパーコンピュータからワークステーションまで、幅広い環境で利用されることを想定している。更にこれら様々な環境で、ある程度のパフォーマンスを提供することを目的として開発を行っている。GPU および TCA を搭載した超並列大型計算機の典型的な環境である HA-PACS 及びメニーコア環境である COMA 及び OFP 上で具体的な動作検証ができ、格子 QCD の本格計算に貢献できた。

4. 今後の展望

今後のペタスケールからエクサスケールへと期待される計算機は、オーダー10,000 を超えるメニーコアの環境となっていくと考えられる。本申請課題によるメニーコア計算機向けコード開発をさらに押し進め、エクサスケール時代でも十分適用可能なコードを目指す。

5. 成果発表

(1) 学術論文

(2) 学会発表

金森逸作「格子 QCD フェルミオンの SIMD 機向けの実装：KNL 機でのドメインウォールフェルミオン」日本物理学会第72回年次大会, 2017年3月17日～20日, 大阪大学. (OFP 試験運用期間における OFP 利用を中心とした内容を発表)

(3) その他

使用計算機	使用計算機に○	配分リソース*
HA-PACS	○	180
HA-PACS/TCA	○	90
COMA	○	67.5
※配分リソースについては 32node 換算時間をご記入ください。		

