

研究責任者名 Name	上田 悟 Satoru Ueda	所属機関 Affiliation	高エネルギー加速器研究機構 素粒子原子核研究所
受理番号 Proposal No.	大型 13/14-19	研究課題名 Program title	格子 QCD 共通コードの開発と高速なアルゴリズムの研究

研究を終了しましたので、下記の通り報告します。

成果の概要

**Abstract**

(和文)

オブジェクト指向に基づいた C++言語による格子 QCD コード Bridge++の開発を行い、  
2014 年 9 月 19 日に version 1.2 に更新し公開した。現在も精力的な開発が続いている。

version1.2 では

- OpenMP による thread 並列化対応
- BG/Q への対応とチューニング

が主な特徴である。

(英文)

We developed the lattice QCD code Bridge++ using C++ language which allows an Object oriented programming and upgrade to version 1.2 at September 19th, 2014. We are developing continuously.

In this term of research, Main feature of version 1.2 is

- thread parallelization by OpenMP,
- made the code compatible with BG/Q and tuning.

研究成果を公開しているホームページアドレス

<http://bridge.kek.jp/Lattice-code/>

研究成果の 公表	口頭研究発表 件数	査読つきの 学術論文数	プロシーディング 論文数	その他 (投稿中を含む)
	2	0	1	1

成果の公表リスト（それぞれの枠に番号をつけて記入願います。）

口頭研究発表 Presentations at scientific meetings concerning the program		
1. International Conference on Computational Science (ICCS 2014), 10-12 June, Cairns, Australia 2. The 32nd International Symposium on Lattice Field Theory (Lattice 2014) 23-28 June, 2014, Columbia University, New York		
査読付きの学術論文(雑誌名等には 巻、頁、発表年を記載) (*) 不足する場合には追加願います。 Refereed Journal Articles (name of journal, volume, page, year)		
1	著者名 Author	
	タイトル title	
	雑誌名 name of journal	
	URL	
2	著者名	
	タイトル	
	雑誌名等	
	URL	
3	著者名	
	タイトル	
	雑誌名等	
	URL	
プロシーディング論文(雑誌名等には 巻、頁、発表年を記載) (*) 不足する場合には追加願います。 International Conference Proceedings (name of journal, volume, page, year)		
1.	著者名 Author	S. Motoki et al.
	タイトル title	Development of Lattice QCD Simulation Code Set “Bridge++” on Accelerators
	雑誌名等 name of journal	proceedings of 2014 International Conference on Computational Science
	URL	<a href="http://www.sciencedirect.com/science/article/pii/S1877050914003329">http://www.sciencedirect.com/science/article/pii/S1877050914003329</a>
2.	著者名	
	タイトル	
	雑誌名等	
	URL	
3.	著者名	
	タイトル	
	雑誌名等	
	URL	
その他 (学位論文、紀要、投稿中の論文を含む) (著者、タイトル、論文種別、URL を記載) Others (thesis for a degree, bulletin, papers to be published, etc.)		
S Ueda et al, “Lattice QCD code Bridge++ on multi-thread and many core accelerators”, proceedings of the 32nd International Symposium on Lattice Field Theory		
特記 (本研究に関係した、新聞記事・著作、受賞など) (過去に遡っても構いません。) Special Notes (newspaper article, literary works, awards, etc.)		

# 「格子 QCD 共通コードの開発と高速なアルゴリズムの研究」実施報告書

2014 年 12 月 01 日

課題グループ名: scbridge

研究責任者: 上田悟(sueda@post.kek.jp)

## 研究組織

以下の人員(氏名・所属・職名)と役割分担で研究を行った。

- 上田 悟 高エネルギー加速器研究機構 素粒子原子核研究所 研究員  
責任者: 全体統括
- 松古栄夫 高エネルギー加速器研究機構 計算科学センター 助教  
役割分担: 数値計算アルゴリズムの開発・応用、コードの開発・チューニング
- 元木伸治 高エネルギー加速器研究機構 素粒子原子核研究所 研究員  
役割分担: 数値計算アルゴリズムの開発・応用、コードの開発・チューニング
- 滑川裕介 筑波大学 計算科学研究センター 研究員  
役割分担: 数値計算アルゴリズムの開発・応用、コードの開発・チューニング
- 石原 誠 理化学研究所 その他  
役割分担: 数値計算アルゴリズムの開発・応用、コードの開発・チューニング
- 中尾昌広 理化学研究所 研究員  
役割分担: 数値計算アルゴリズムの開発・応用、コードの開発・チューニング
- 村井 均 理化学研究所 研究員  
役割分担: 数値計算アルゴリズムの開発・応用、コードの開発・チューニング

## 格子 QCD 共通コード Bridge++の開発

### 1. 概要

本研究における成果の中心は格子 QCD シミュレーション共通コードの開発にある。我々はオブジェクト指向に基づいた C++言語による格子 QCD コード Bridge++ の開発を行った[1]。この開発プロジェクトは平成 20 年度科研費・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」(領域代表・青木慎也、筑波大学)において開始し、その後、HPCI 戦略プログラム 分野5「物質と宇宙の起源と構造」のサポートも受けている。様々な格子作用やアルゴリズムを適用可能で、ノート PC から超並列計算機まで幅広いアーキテクチャに対応し、最先端の研究に必要なパフォーマンスを実現でき、なおかつ使い易いものを目指している。2012年7月24日に version 1.0 を公開し、本研究期間の2014年9月19日に version1.2 に更新した。現在も精力的な開発が続いている。

本研究期間で公開した version1.2 の特徴は、

- OpenMP による thread 並列化対応
- Blue Gene/Q への対応とチューニング

である。さらに、コードの更新する際に並列化及びコンパイラ依存性の検証にも、本システムを利用した。GNU C++, Intel C++と異なるコンパイラとハードウェアによる検証が可能であり、また実践規模のシミュレーションを容易に実現する環境として、KEKSC システムは Bridge++の開発に不可欠な役割を果たしている。

本研究期間には国際会議で本コードの紹介を 2 回行った。

また、指示文による並列プログラミング言語である XcalableMP によるコード開発を行い、可読性や実行性能を確かめ XMP が MPI の代わりとして利用できるかどうかを確かめるために、XcalableMP の移植作業を行った。

### 2. OpenMP による thread 並列化 及び BG/Q へのチューニング

前研究期間の報告にて BG/Q 32 nodes に対する性能を 13%としていたが、Flops カウンターのバグにより正しく評価できていなかった。そのため、計算量を手で計算し、実行時間から計算したところ格子 QCD 計算の大部分を占めるフェルミオン演算子の計算で 5,6%であった。

本研究期間では OpenMP による thread 並列化を含め最適化を進めた結果、一番簡単な Wilson 型作用の計算で 14%程度、より複雑な Clover 型作用で 8%程度まで最適化することができた。この成果は version1.2 として公開済みである。

今後、さらなる最適化をすすめ HMC による配位生成全体で 10%程度の実行速度を達成できるよう開発を進める予定である。

今後の主な課題は

- OpenMP による thread 並列を行う部分の拡大

- BG/Q の `intrinsic` の活用
- `Wilson library` の活用

などである。

### 3. XcalableMP の移植

#### 3.1 概要

XcalableMP (XMP)は、PC クラスタコンソーシアム XMP 規格部会が提案する並列プログラミング言語である。XMP を用いることで、デファクト標準となっている MPI よりも容易に並列プログラムを記述することができる。

本作業は、XMP コンパイラ `Omni XcalableMP` を、日立 SR および IBM BlueGene へ移植することを目的とする。これにより、格子 QCD シミュレーションの分野においても並列プログラム開発に関する生産性を向上できる。

#### 3.2 作業

`Omni XMP` は、大きく次の 3 つから成る。

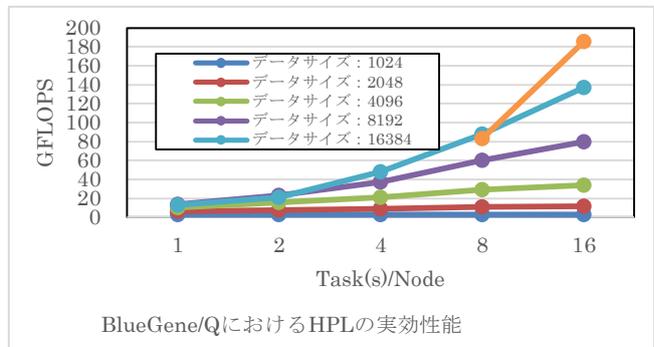
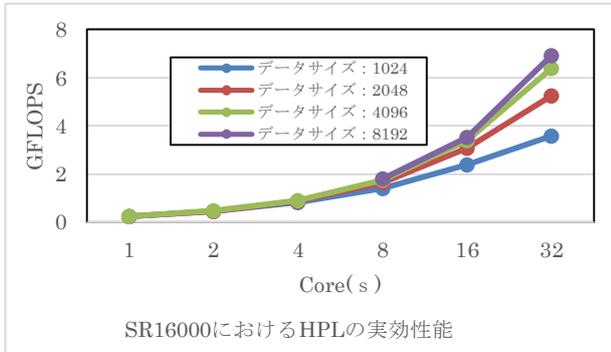
- ドライバ: コンパイルオプションを処理し、トランスレータおよびバックエンドクロスコンパイラを適切なオプションで呼び出す。
- トランスレータ: XMP ソースプログラムを、MPI ソースプログラムへ変換する。
- 実行時ライブラリ: 変換後 MPI ソースプログラムとリンクされ、各種の実行時処理を行う。

SR16000 の計算ノードでは AIX が、BlueGene/Q の計算ノードでは独自 OS が、それぞれ動作している。また、SR16000 のフロントエンドでは、AIX が動作している。そのため、これまで Linux で動作することを想定して開発されてきた `Omni XMP` を移植するためには、多くの試行錯誤が必要であった。いくつかの例を以下に示す。

- SR16000 上で提供される GNU ユーティリティ群 (`bison` 等)の挙動が安定せず、自前でインストールする必要があるがあった。
- SR16000 のフロントエンド上で動作するトランスレータを `gcc` でコンパイルする際に、`-maix64` オプションを指定する必要があるがあった。
- 両システムのフロントエンドで提供されている IBM 製 Java と、`Omni XMP` で従来使われていた `OpenJDK` に非互換があったため、`configure.ac` を修正した。

#### 3.3 評価

SR16000 および BlueGene/Q の両方において、HPC Challenge ベンチマークおよび姫野ベンチマークの動作を確認し、それらの性能を評価した。結果の一部を以下に示す。



#### 4 参考

[1] [http://bridge.kek.jp/Lattice-code/index\\_j.html](http://bridge.kek.jp/Lattice-code/index_j.html)