

滝沢 寛之

東北大学大学院情報科学研究科
准教授

進化的アプローチによる超並列複合システム向け開発環境の創出

§ 1. 研究実施体制

(1) 滝沢グループ

① 研究代表者: 滝沢 寛之 (東北大学大学院情報科学研究科、准教授)

② 研究項目

- ・システムの複合性を扱うためのプログラミングインタフェースの設計
- ・HPC アプリケーションのためのリファクタリングツールの開発

(2) 高橋グループ

① 主たる共同研究者: 高橋 大介 (筑波大学システム情報系、教授)

② 研究項目

- ・超並列複合システムに向けた高速フーリエ変換 (FFT) ライブラリおよび代数的多重格子法ライブラリの開発
- ・混合精度計算に対応した基本線形計算ライブラリの基本設計

(3) 須田グループ

① 主たる共同研究者: 須田 礼仁 (東京大学情報理工学系研究科、教授)

② 研究項目

- ・ドメイン特化型ツールによる超並列複合システムの階層的抽象化に関する研究

(4) 江川グループ

① 主たる共同研究者: 江川 隆輔 (東北大学サイバーサイエンスセンター、准教授)

② 研究項目

- ・HPC のためのリファクタリングカタログの設計

- ・超並列複合システム向けのアプリケーション開発手法の確立

§ 2. 研究実施の概要

現在、多くの高性能計算アプリケーションは特定のシステムを想定し、その想定システム上で高性能を達成できるように開発・最適化されている。このため、そのアプリケーションを他のシステム上で実行する場合には、高い性能を達成できない恐れがある。すなわち、性能可搬性が低いことが問題となっており、既存のアプリケーション資産を新システムに移行する場合にも同様の問題が発生する。このため本研究では、既存のアプリケーション資産の性能可搬性を高めることで、その資産の将来のシステムへの円滑な移行を実現する。そのためには性能可搬性の阻害要因を大規模なアプリケーションコードから分離する必要がある、書き直しの困難な規模のアプリケーション資産を対象に考える場合には、コードを段階的に再構成することで分離を実現する必要がある。この分離作業を、本研究では HPC リファクタリングと呼ぶ。

本研究の大きな目標の一つは、事例に基づいて性能可搬性の阻害要因を体系的にまとめ、それらを切り離す HPC リファクタリングの方法論をまとめることである。平成 26 年度には、前年度に引き続き現在東北大学サイバーサイエンスセンターで実行されているシミュレーションコードの最適化、並列化を通して事例集の収集に取り組み、ベクトル機向けに施されたループレベル最適化事例のさらなる収集と、地球シミュレータセンター等、他の計算機センターで実行されているスカラ機、アクセラレータを視野にいたした事例集の拡充に取り組んだ。併せて、これらの事例集における最適化前後のコードを複数の計算環境で実行可能なカーネル群として整備を進めた。

また、HPC リファクタリングによってアプリケーションコードから切り離された情報は、特定のシステムで性能を出すための情報と言い換えることができ、単に切り離すだけでは性能が低下する。このため、各システムで性能を出すための情報をアプリケーションコードとは別の存在として抽象化し、システムに合わせて使い分けることで、それらのシステム間の性能可搬性を実現することを考えている。この抽象化のための最も一般的な方法は、汎用性の高い計算を数値ライブラリとして実装することである。平成 26 年度には、数値計算ライブラリにより超並列複合システムの階層的抽象化を行うために、アプリケーションプログラムをなるべく修正せずに高い性能を達成できるライブラリの研究開発を行った。また、後述の Xevolver ディレクティブの数値計算ライブラリへの適用について検討を行った。超並列複合システムに向けた高速フーリエ変換 (FFT) ライブラリ、代数的多重格子法ライブラリ、そして混合精度計算に対応した基本線形計算ライブラリの研究開発を行った。さらには、ポストペタ時代の計算科学で必要性が明確化されてきた、新世代アルゴリズムの研究も進めている。超大規模計算で主要なオーバーヘッドとなる通信を削減したアルゴリズムとして、チェビシェフ基底 CG 法、ブロックチェビシェフ基底 CG 法、通信削減型正規直交規定生成法、また、数万ノード以上で急速に課題となる OS ジッタの影響を回避する集団通信アルゴリズムの開発を行った。

上記のように抽象化する仕組みを用意しても、全てを抽象化することはできないため、アプリケーションコードの修正が求められる。そのようなコード修正は一般化が難しく、特定のシステムやアプリケーション向けの修正となる傾向にある。そのようなコード変換も何らかの形でアプリケーションコードから切り離し、再適用できる仕組みがなければ、HPC リファクタリングされたコードで元のコードと同じ性能を出すことはできない。このため本研究では、そのようなシステム特有、アプリケーション特有のコード修正を、ユーザ定義変換として定義するための仕組みを構築してきた。平成 26 年度は、

構築した **Xevolver** フレームワークをいくつかの実アプリケーションに適用し、複数のシステムで高い性能を達成できる、すなわち高い性能可搬性を実現できることを明らかにした。現在、**Xevolver** フレームワークを公開し、チーム内外の共同研究者からのフィードバックを集めている。また、コンパイラの内部表現である抽象構文木に対する変換規則を直接記述する低水準インターフェースに加えて、高水準インターフェースの可能性について検討した。高水準インターフェースでは、ディレクティブを含む **Fortran** のダミーコードで変換を記述する。また、変換操作はスクリプト言語を通じて拡張可能である。

【代表的な原著論文】

- [1] Ryusuke Egawa, Kazuhiko Komatsu, Hiroaki Kobayashi, "Designing an HPC Refactoring Catalog Toward the Exa-scale Computing Era," Sustained Simulation Performance 2014, pp 91-98, 2014.
- [2] Akihiro Fujii and Osni Marques, "Axis Communication Method for Algebraic Multigrid Solver," IEICE Transactions on Information and Systems, Vol.E97-D, No.11, pp. 2955-2958, 2014. (DOI: 10.1587/transinf.2014EDL8052)
- [3] Hiroyuki Takizawa and Shoichi Hirasawa and Yasuharu Hayashi and Ryusuke Egawa and Hiroaki Kobayashi, "Xevolver: An XML-based Code Translation Framework for Supporting HPC Application Migration," The 21st annual IEEE International Conference on High Performance Computing (HiPC 2014), 2014.