

「ポストペタスケール高性能計算に資するシステムソフトウェア技術の創出」
平成 23 年度採択研究代表者

H27 年度
実績報告書

千葉 滋

国立大学法人 東京大学 情報理工学系研究科
教授

ポストペタスケール時代のスーパーコンピューティング向けソフトウェア開発環境

§ 1. 研究実施体制

(1)「千葉」グループ

- ① 研究代表者:千葉 滋 (東京大学大学院情報理工学系研究科、教授)
- ② 研究項目
 - ・スーパーコンピューティングのためのモジュール機構

(2)「増原」グループ

- ① 主たる共同研究者:増原 英彦 (東京工業大学大学院情報理工学研究科、教授)
- ② 研究項目
 - ・スーパーコンピューティングのための動的言語処理系

(3)「鶴林」グループ

- ① 主たる共同研究者:鶴林 尚靖 (九州大学大学院システム情報科学研究院、教授)
- ② 研究項目
 - ・スーパーコンピューティングのための検証技術

(4)「五十嵐」グループ

- ① 主たる共同研究者:五十嵐 淳 (京都大学大学院情報学研究科、教授)
- ② 研究項目
 - ・スーパーコンピューティングのための型システム

§ 2. 研究実施の概要

GPUなどを搭載した異種混在型のスーパーコンピュータが既に一般的になっている今日、ポストペタスケール時代のスーパーコンピューティング環境では、このようなハードウェアを駆使したプログラムの開発を支援する様々な開発ツール、ライブラリ、言語が求められている。実際、プログラミング言語については、従来の Fortran/C だけでなく、C++ あるいは Julia や Python のようなより抽象度の高い言語が広く使われるようになる兆しもある。そのようなプログラムは従来以上に複雑であり、支援ツールなしに手作業だけでプログラムを作成すると、開発効率が著しく悪くなる。特に抽象度が相対的に低い従来型の言語を用いる場合、その傾向は大きい。機械的・自動的な開発支援ツール等なしでの手作業に頼った開発はもはや現実的ではない。我々はそのような目的意識からソフトウェア技術の研究開発をおこなってきた。

今年度の主な成果は、GPU を用いた並列プログラムの正しさを自動検証するための理論的基盤を構築したことである(文献1)。プログラムの自動検証とは、人間が書いたプログラムが果たして意図通りの計算を正しく実行するのか、メモリ競合のような誤りを持たないか、機械的に検査することである。並列プログラムは従来型の逐次プログラムよりも複雑なため、そのような自動検証の技術が強く望まれている。我々はそのような検証を実現するための論理体系を定理証明支援器 Coq を用いて形式化し、機械的な検証を実現する土台を構築した。

もう一つの成果は、やはり並列プログラムの正しさを機械的に検証するための技術の研究である(文献2)。前者の成果はプログラムを実行する前に機械的に正しさを検査するための技術のためのものであるが、こちらの成果はプログラムの実行後に実行が正しかったのかを検査するための技術である。両者はそれぞれ静的検証、動的検証と呼ばれ、互いに補完する関係にある。大規模な並列分散プログラムでは、性能向上のため、個々の計算順序を手作業で入れ替えるが、この入れ替えによって一部の計算が実行されなくなっていないか、あるいは二重に実行されていないか、を実行後に確認するシステムを開発した。このシステムを用いると、ドメイン専用言語を用いて検査したい計算の内容を記述するだけで、検査対象のプログラムの中に検査のためのコードがシステムによって自動的に埋め込まれ、実行後に検査結果がレポートされる。このシステムを用いて実際に既存のプログラムの誤りを発見し、有用性を実証した。

【代表的な原著論文】

- (1) Izumi Asakura, Hidehiko Masuhara, and Tomoyuki Aotani. "Proof of soundness of concurrent separation logic for GPGPU in Coq." *Journal of Information Processing*, Vol. 23, No. 1, pp. 132-140, January 2016.
- (2) Yoshiki Sato, Shumpei Hozumi and Shigeru Chiba, Calculation Coverage Testing for Scientific Applications, *International Symp. on Software Testing and Analysis (ISSTA'15)*, ACM, pp.350-360, 2015.