

千葉 滋

東京大学大学院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

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(あるいはハイパフォーマンスコンピューティング)では、ハードウェアも高度に進歩するが、もはやソフトウェア側もハードウェアの性能を引き出すためには相応の負担が必要となる。一方で、従来型の科学技術計算だけでなく、生命科学分野や社会科学分野などのアプリケーションソフトウェアもスーパーコンピューティングを利用するようになることが予期され、利用の裾野を広げるためには、一般利用者のプログラミングの負担を下げるソフトウェア技術の研究開発が急務である。

我々は、旧来型の Fortran や C 言語ではなく、Java や Ruby 言語のような、比較的現代的でソフトウェアの開発生産性が高いとして広く用いられているが、スーパーコンピューティングに用いるには実行時性能が十分でないと言われている言語をスーパーコンピューティングに利用可能にする技術を研究開発している。これに加えて我々は、そのような実行基盤の上に構築するフレームワークないしはドメイン専用言語の実現のために、ソフトウェアの先進的なモジュール化技術も合わせて研究した [1]。またスーパーコンピューティングの新しい応用分野として、ソフトウェアのリポジトリマイニングや社会シミュレーションの並列処理についても研究している。特に、ソフトウェア工学分野の研究においてもスーパーコンピュータの活用でより優れた成果が生み出せることを実証することを目指している。さらに HPC プログラムの開発を支援するために、プログラムの正しさの検証技術の研究に取り組んでいる。

平成 26 年度は、Java 言語ベースの処理系として Bytespresso の開発を開始した。前年度までに開発した処理系 WootinJ での知見を元により実用性の高い処理系をめざした開発と、そのための基礎技術の研究をおこなった [2]。この処理系は devirtualization、inlining などの技術を駆使して、オブジェクト指向で書かれたコードを効率の良い CUDA コード等に変換する技法を実現している。さらにこのような処理系の上にライブラリとして実現された DSL の構文の自由度を改善する研究をおこなった [3]。また Ruby 言語ベースの処理系として Ikura の開発を継続しておこなった。今年度は特に、現実的なステンシル計算を記述するための API 設計、soft-typing 方式に基づく型システムとその推論系、複数のバックエンドをサポートするコード生成系に注力して開発を進めた。またソフトウェアのリポジトリマイニングの並列処理については、部分的な変形を含むコードクローン検出(プログラム中の重複コードの発見)を例題に、実際にスーパーコンピュータを利用して大規模ソフトウェアリポジトリに対する実行を試みた。ソフトウェアリポジトリとは、複数のソフトウェアプロジェクトのソースコードの集合体である。社会シミュレーションの並列処理に関しては、同じ研究領域の野田チームと研究協力を進め、同チームが持つ社会シミュレーションのソフトウェアを題材に、エージェントシミュレーションを効率よく並列実行するための技法を研究した。HPC プログラムの検証技術については、ホーア論理に基づいた検証系の理論基盤を構築し、CUDA を対象に実際に動作する検証系の開発に着手した。

【代表的な原著論文】

- [1] Tetsuo Kamina, Tomoyuki Aotani, and Hidehiko Masuhara. Generalized layer activation mechanism through contexts and subscribers. In Proc. of International

Conference on Modularity (Modularity'15), pp.14-28, March 2015.
DOI:10.1145/2724525.2724570

- [2] Maximilian Scherr and Shigeru Chiba, “Implicit Staging of EDSL Expressions: A Bridge between Shallow and Deep Embedding,” In Proc. of the 28th European Conference on Object Oriented Programming (ECOOP 2014), LNCS 8586, Springer, pp. 385-410, 2014. DOI: 10.1007/978-3-662-44202-9_16
- [3] Kazuhiro Ichikawa and Shigeru Chiba, “Composable User-Defined Operators That Can Express User-Defined Literals,” In Proc. of the 13th international conference on Modularity (MODULARITY '14), pp. 13-24, April 2014. DOI: 10.1145/2577080.2577092