

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 (東京大学大学院総合文化研究科、准教授)
- 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. 研究実施内容

千葉グループは、昨年度より開始した GPGPU を搭載した多数の分散ノードから構成されるスーパーコンピュータ向けの Java 言語実行系 WootinJ の開発を引き続き行った。これは Java 言語をコンパイルした結果得られる Java バイトコードから、C/CUDA プログラムへ実行時に得られる型情報などを用いて実行時にコード変換をおこなうシステムである。同時に Java バイトコードに埋め込まれた情報に基づいて CUDA や MPI の命令を変換結果のコードに埋め込む。これにより Java 言語のクラスやオブジェクトを活用した並列プログラミング・フレームワークを作り、それを使ってアプリケーション・プログラムを開発しても、モジュラリティや再利用性を考慮せずに書かれた従来型のプログラムに匹敵する性能で実行できるようにすることをねらっている。今年度はコード変換系の実装を進めるとともに、拡散方程式を実装するためのフレームワークを応用例として開発に取り組んだ。これを用いると Java で汎用的に書かれた主プログラムは一切変更せずに、利用するオブジェクトのクラスを変えるだけで、CUDA や MPI 向け、あるいは両方を融合したコードを自動的に生成できるようになる。性能測定にあたっては東京工業大学の TSUBAME2.0 を利用している。また HPC ライブラリ向けのプロダクトライン化に向けて言語機構の研究を進め、アスペクト指向とイベント指向を融合した言語機構を提案した[1]。さらに構文拡張の柔軟性を高め、より直感的な DSL (Domain Specific Language) をフレームワークあるいは内部 DSL として実装できるようにするための基礎技術の研究を継続しておこなった。昨年度に試作した言語処理系 LasticJ の後継版である ProteaJ を開発し、鶴林グループと共同して有用性の評価をおこなった。

これらの研究活動に加え、千葉グループでは東京大学情報基盤センターのスーパーコンピュータである富士通 FX10 上での研究の準備として、Java 仮想機械 OpenJDK の FX10 向けの移植作業をおこなった。また並列言語 X10 の FX10 への移植も並行して実施した。この成果は同じ研究領域の「超大並列計算機による社会現象シミュレーションの管理・実行フレームワーク(代表: 野田五十樹)」にも提供し、協力して研究をおこなった。

増原グループは高性能計算の最適化機構を提供する土台として Ruby によって GPGPU を行う言語基盤 Ikra の開発を中心に研究を進めた。前年度に着手した型推論系の最初の処理系を完成させ、それを Ikra 変換系に統合した。これにより、基本的なステンシル計算を行う Ruby プログラムから、ほぼ変更なしに型推論した上で CUDA コードを生成することが可能となった。また Ikra の開発と並行して、言語処理系部分のモジュール化のための基礎技術として実行文脈によって挙動を変化させる文脈指向プログラミング言語についての研究を行った。特に事象により文脈を変更する EventCJ 言語とその開発方法論と、五十嵐グループと共同して文脈指向言語の基礎意味論の検討を進めた。

鶴林グループは、高性能計算が可能な DSL 群をソフトウェアプロダクトラインの考え方に基づい

て提供することを研究目標としている。高性能計算を実現するための言語要素(オペレータやデータ型)、計算機能、実装機能などを部品として提供し、それら部品を高性能計算の用途に応じて組み合わせることにより DSL を構築する。部品の組み合わせにおいて矛盾が生じると正しい計算結果が得られなくなる可能性があるため、合成段階において形式検証の技術を導入する。現在、高性能計算の対象ドメインとして、オープンソースソフトウェアを対象としたリポジトリマイニング(大規模データ解析により新たな知見を発見する手法)を想定している。本年度は、フィーチャモデル(部品群とその関係構造をモデル化したもの)に対して DSL として求める要求(機能や実行環境に対する要求)を与えることにより、DSL の文法を導き出す方法を考案した。フィーチャモデルおよび要求(制約)を論理式に変換し、SAT ソルバーにより要求を充足する部品群を抽出する。この実現に Alloy と呼ばれる形式仕様記述言語を用いた。来年度は、DSL の文法だけでなく、処理系も含めて自動化する予定である。具体的には、フィーチャモデルから ArgyleJ(昨年度開発したリポジトリマイニング向け DSL)の自動生成を目指している。自動化に際しては、千葉グループで開発している ProteaJ と WootinJ が重要な役割を果たすことになる。

五十嵐グループは、細粒度にモジュール化されたソフトウェア部品を合成する際の安全性の検証とその効率化についての研究を継続した。特に、文脈指向言語と呼ばれる言語で見られる動的モジュール合成機構の安全性についての理論的考察を行い、モジュール関係の依存関係をプログラム中に宣言することによって、モジュール合成順序の動的変更など、実行時における柔軟性の高いモジュール合成が安全に行えることを示した。並行して、CUDA アーキテクチャなどのいわゆる SIMT 計算モデルにおけるプログラムの正当性を示すためのプログラム論理の構築に着手した。逐次の命令型言語の意味論を Habermaier と Knapp の手法による SIMT 拡張を行い対象言語とし、逐次命令型言語に対するプログラム論理であるホーア論理を拡張し、対象言語の意味論に対する健全性・相対完全性を示すことができた。

§3. 成果発表等

(3-1) 原著論文発表

●論文詳細情報

1. YungYu Zhuang, Shigeru Chiba, "Method Slots: Supporting Methods, Events, and Advices by a Single Language Construct", Modularity:AOSD13, pp.197-208, 2013, DOI 10.1145/2451436.2451460