

「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

|                |
|----------------|
| H23 年度<br>実績報告 |
|----------------|

千葉 滋

東京工業大学大学院情報理工学研究科・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、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型システム

## § 2. 研究実施内容

(文中に番号がある場合は(3-1)に対応する)

千葉グループは、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変換をおこない、それを C/CUDA コンパイラでコンパイル、CPU または GPGPU 上で並列実行する変換・実行エンジンである。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・プログラムを開発することが可能になる。従来の類似システムでも、オブジェクトを使わないような Java プログラムを C/CUDA プログラムに変換して高速並列実行することは可能であったが、WootinJ ではオブジェクトを利用したプログラムも一定の制約の下で高速な C/CUDA プログラムに変換して実行できる。さらに WootinJ の性能を計測するための実験を東京工業大学の TSUBAME2.0 上で開始し、現在のところ良好な性能を得ている。

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開始した。いくつかのアイデアを考案し、その有用性を調べるための処理系のプロトタイプを Java 言語をもとに試作した。試作した言語処理系 LasticJ は、千葉グループ内で有用性の評価を進めるとともに、鶴林グループにも提供し評価を進めている。

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。Ikra は以前から試作を進めてきたものであるが、研究開始以降には並列データクラスの再設計および型推論系の核部分の設計と構築を行った [6]。設計した並列データクラスは、土台となる Ruby 言語の配列クラスを多次元に拡張し、典型的な数値計算等の応用を素直に記述できるようにした。型推論系の核部分は、GPU 上で実行可能な並列核部分のみを型付けすることを一次目標として、単相型推論系にメソッドとクラスを導入した単純なものとした。これによって初歩的なデータ並列計算を行う Ruby プログラムを人手による型注釈を付加することなく GPU へコンパイル・実行することが可能になり、次年以降に予定している複雑なデータアクセスパターンを持つプログラムの検討および最適化機構の検討を行う準備が整った。また、並行して最適化機構のためのモジュール化のために有用となる技術としてアスペクト指向プログラミングにおける汎化型サポートの強化手法を提案した[5]。

鶴林グループは、リポジトリマイニングを対象とした「高性能計算が可能な DSL」を提供することを研究目標としている。この DSL はソフトウェアプロダクトラインに基づいて言語部品を検証つきで合成することにより構築することを狙っている。リポジトリマイニングとは、大規模データ解析により新たな知見を発見する手法であり、高性能計算技術の応用例の一つである。

本年度はまず、ソフトウェアリポジトリマイニングを対象にフィーチャ分析(ソフトウェアプロダクトラインを構築するためのドメイン分析手法)を行い、必要となる言語部品群を洗い出した。また、言語部品の組み合わせが適切か否かの検証はフィーチャモデル上の制約を充足するか否かにより判定することとした。DSLのプロトタイプは内部DSLと外部DSLの2種類を開発した。前者は千葉グループで開発したLasticJを用いて開発した[3]。Javaをホスト言語とするDSLであり、Java上で容易にリポジトリマイニング用のアプリケーションを開発できる。後者はXtextと呼ばれるDSL構築ツールを用いて開発した。リポジトリマイニング専用の言語であり、CUDAコードに変換される。平成23年度のプロトタイプ開発を通じて、1)言語部品の構築にあたっては更に深いドメイン分析が必要、2)汎用的な検証機構だけでは十分でなくドメインに特化した検証規則やアルゴリズムが必要、ということが明らかになった。

五十嵐グループは、千葉、五十嵐、Zakilovによって提案・実装されたプログラミング言語であり、本研究においてプロダクトライン・アーキテクチャ記述のための言語として拡張・開発されるGluonJ言語上でのジェネリクスおよび型特化機構についての型システム的设计と実現可能性の理論的検討を行った。

検討の結果、型システムの観点からはジェネリクスとGluonJ特有の言語機構はほぼ直交しており、両者の計算モデルを融合することにより型安全性は保証できそうである、という結論に達したが、機能実現の観点からは、現在のJavaジェネリクスの実装を大きく変更する必要があることがわかった。特に、ジェネリクスの機能から自然に得られる型特化機構を実現するためには、プログラム実行時にジェネリクスの型変数の値(実行時型情報)を必要となり、素朴に実行するとオーバーヘッドが大きいことが予想される。このオーバーヘッド削減のための効率的な実装のための機能制限を検討する必要があるという中間的結論を得た。

## § 3. 成果発表等

### (3-1) 原著論文発表

#### ●論文詳細情報

[1] Shumpei Akai and Shigeru Chiba, “Method Shelters: Avoiding Conflicts among Class Extensions Caused by Local Rebinding,” MODULARITY: aosd 2012, ACM, 2012 (accepted).

### (3-2) 知財出願

① 平成 23 年度特許出願件数(国内 0 件)

② CREST 研究期間累積件数(国内 0 件)