

ルビーによる高生産な超並列・超分散計算ソフトウェア基盤 Highly productive software environment based on Ruby for parallel and distributed computing systems

平木 敬 (HIRAKI KEI)

東京大学・大学院情報理工学系研究科・教授



研究の概要

関数型オブジェクト指向言語である Ruby を拡張し、1 プロセッサから 10 万プロセッサ規模までスケールアップである HPC 向け高生産言語として確立する。Ruby の特徴である Web 環境と計算環境の統合を生かし、HPC 情報環境における Web インタフェースと、計算を融合するシステムを開発し、超並列・超分散透明性を持つソフトウェア体系を実現する。

研究分野：総合領域

科研費の分科・細目：情報学 ・ 計算機システム・ネットワーク

キーワード：HPC、高生産性言語、超並列計算、最適化コンパイラ

1. 研究開始当初の背景

言語システムによる高生産性の実現は、米国 HPCS プロジェクトにおいて試みられた。しかしながら HPCS 言語である X10, Fortress, Chapel は、広く現場の研究者・開発者に普及する可能性、10 万を超す並列性を持つ超並列スーパーコンピュータへのスケラビリティにおいて、多くの問題点を持つ。(1)利用者層が限定されるため、ユーザ層の広がり、開発者の広がりが限定されること、(2)コンピュータサイエンスが専門でない研究者・開発者にとって使用経験がない数値計算専用新言語は、習得意欲が低く、普及が困難となること、(3)オブジェクト指向言語は、一般的な意味でプログラムの生産性を向上させるが、各種並列性の記述と、過大な並列性を避けるための並列性制御の問題について解決策とはならない。

2. 研究の目的

本研究開発では関数性とオブジェクト指向性を持ち、柔軟性を持つとともに、広く普及している Ruby 言語をベースとし、それを超並列化、数値計算向けに拡張し、上記問題点を根本的に解決することを目的とする。

Ruby の持つ関数性やオブジェクト指向性に加え、動的プログラミングや他言語・外部接続の柔軟性を生かしたまま、スーパーコンピュータプログラミングの生産性を著しく向上させ、スケラビリティを実現する。また、本研究開発では、Ruby を用いた問題定式

化の題材として、理論天文学シミュレーション、惑星シミュレーション、分子動力学、並列探索問題という、日本が世界的に見て優れている分野を選択し科学的に意義のあるアプリケーションソフトウェアの Ruby による定式化を通して HPC Ruby の記述力を示す。

3. 研究の方法

5 年間の研究開発で関数型オブジェクト指向言語である Ruby を HPC 向け高生産言語として確立する。また、Ruby の特徴である Web 環境と計算環境の統合を生かし、HPC 環境における Web インタフェースと、計算を融合するシステムを開発し、超並列・超分散透明性を持つソフトウェア体系を実現する。これらの目的を実現するため、(1)Ruby 言語 HPC 最適化に関する研究、(2)Ruby を用いた問題定式化に関する研究、および(3)超分散・超並列計算環境による評価の 3 個のサブテーマに分かれて研究を実施する。

具体的には、(1)Ruby プログラムから並列性を抽出するコンパイル方式、動的言語における最適化コンパイル技法、特に繰り返し構造および多次元データ処理の最適化技法、過大な並列性による性能低下を防止する並列性制御方式を実現する。並列分散処理の動的状況により最適化を繰り返す適応型コンパイラの研究開発を実施する。(2)では、Ruby の持つ動的性質を活用するシミュレーションコードを開発し、情報科学を専門としない研究者が容易に分散並列シミュレーション

ソフトウェアを作成できる基盤を構築する。(3)では、評価システムの整備し、Webシステムとの統合による Rubyらしい科学技術計算の実現と評価を実施する。

4. これまでの成果

平成 23 年度までに、シングルプロセッサを対象とした HPC Ruby スタティック・コンパイラの開発を完了し、評価を行った。開発した HPC Ruby コンパイラは部分評価・抽象解釈を組み合わせた、静的型解析・関数束縛解析手法を用い、Ruby 言語で記述されたソースプログラムの変数・メソッド解析と型解析を行い、安全な範囲でのプログラムの静的変換を可能とするものである。

Ruby の特徴である動的実行構造が実際に実行されると、ランタイムに検出しプログラムの逆最適化と Ruby インタプリターの呼び出しが行われる。この方法を用いることにより対象言語の仕様に拡張・制限を課さないという大きな特徴を実現した。

具体的には、プログラムが Eval などの動的構造を実行すると、実行を検出するコードが埋め込まれていることにより、安全に実行がインタプリターに渡され、実行を続ける。

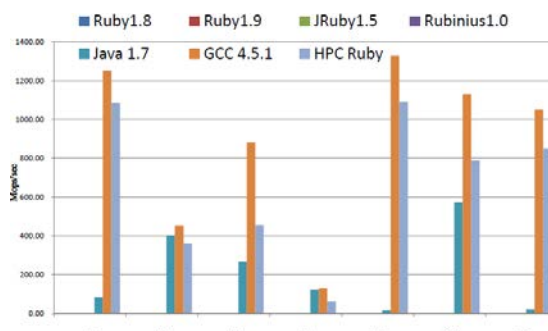
開発したコンパイラに組み込んだ最適化技術は以下の通りである：

1. Method Lookup Elimination
2. Interval Analysis
3. Primitive Operation Inlining
4. Unboxing of Floating Point Numbers
5. Object Recycling
6. Object Flattening
7. Auto Parallelization of Iteration

開発したコンパイラを浮動小数点演算中心のプログラムで評価した。評価は x86 シングルプロセッサで行った。図 1 は浮動小数点演算中心のプログラム例として NAS Parallel Benchmarks (NPB) を実行し、C、Java および既存の Ruby 実行システム (Ruby1.8 および 1.9) の性能と比較した結果である。Ruby のコードは Java 版 NAS Parallel Benchmarks をトランスレータにより自動的に Ruby に変換したもので、人手による最適化は行っていない。

図 1 の評価結果では、NPB に含まれるどのベンチマークでも安定して C の 50% から 80% の性能を得られることが示された。また、Java との比較では、IS では Java が優位であり、CG で同程度であることを除き、Java に対する優位性が示された。なお、従来の Ruby 処理系がグラフ上で見えないが、Ruby1.8 と比較して 800 倍、1.9 に対して 400 倍程度の高速化が実現し、グラフ上で見えなくなっているためである。C との性能差の原因は、配列および多次元配列構造の差によるデータアクセスオーバーヘッド、動的プログラム実行検出のための埋め込みコードのオーバー

ヘッドが中心である。



本研究開発のその他の成果として、天体シミュレーション・プログラムの Ruby による定式化と記述、分散並列 Ruby 実行系実現のためのクラスタコンピュータ整備を行った。

5. 今後の計画

HPC Ruby 処理系の研究開発では、分散並列処理および超並列処理を実現する自動並列化コンパイラの開発を行う予定である。自動並列化を高効率に実現することで、我々の目標である C、Fortran と MPI で記述する場合より高速な計算の実現を試みる。また多くの科学技術シミュレーションを Ruby で記述し、HPC Ruby の優位性を示す予定である。さらに、超高速ネットワーク技術と超並列プロセッサ技術を融合させた HPC Ruby 分散並列システムを実現し、実際に多くの科学者からアクセスできるシステムを構築する予定である。

6. これまでの発表論文等 (受賞等も含む)

・Naoki Tanida, Mary Inaba, Kei Hiraki, “Adaptive auto-tuning of TCP pacing”, The Fifth International Workshop on Automatic Performance Tuning, iWAPT2010, USB メモリ, 2010

・芝哲史, 笹田耕一, 卜部昌平, 松本行弘, 稲葉真理, 平木敬: 実用的な Ruby 用 AOT コンパイラ, 情報処理学会論文誌 (PRO), vol. 4, no. 1, pp. 90-108, 2011

・Junichiro Makino, Hiroshi Daisaka, Toshiyuki Fukushima, Yutaka Sugawara, Mary Iaba, Kei Hiraki, “The performance of GRAPE-DR for dense matrix operations”, Procedia Computer Science vol. 4, pp. 888- 897, 2011

・Hisanobu Tomari, Kei Hiraki, “Retrospective Study of Performance and Power Consumption of Computer Systems”, 情報処理学会論文誌 (ACS), vol. 4, no. 4, pp. 1-11, 2011

・笹田耕一, 卜部昌平, 松本行弘, 平木敬, 「Ruby 用マルチ仮想マシンによる並列処理の実現」, 情報処理学会論文誌 (PRO), vol. 5, no. 2, pp. 25-42, 2012

ホームページ等

<http://data-reservoir.adm.s.u-tokyo.ac.jp/hiraki@is.s.u-tokyo.ac.jp>