

**科学研究費助成事業 研究成果報告書**

平成 27 年 6 月 5 日現在

機関番号：12601

研究種目：基盤研究(S)

研究期間：2009～2014

課題番号：21220001

研究課題名(和文) 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

研究課題名(英文) Highly productive software environment based on Ruby for parallel and distributed computing systems

研究代表者

平木 敬(Hiraki, Kei)

東京大学・情報理工学(系)研究科・教授

研究者番号：20238348

交付決定額(研究期間全体)：(直接経費) 166,600,000円

研究成果の概要(和文)：本研究開発では、関数型オブジェクト指向言語であるRubyを拡張し、HPC向け高生産言語としてHPC Ruby言語を確立した。また、Rubyの特徴である計算環境の統合を生かし、HPC情報環境における新しいソフトウェア体系を実現した。HPC向け新言語の普及のため地球科学分野、天文分野、離散最適化分野においてRuby言語モデルを用いて問題定式化し、Rubyの科学技術計算位における優位性を示した。

分散実行環境の実証研究では、日米欧を100Gbpsインターネットで結び、その90%を高効率利用する通信方式を確立し、実験により実証した。これらの成果を総合し、Rubyを中心とした科学技術計算の体系を確立した。

研究成果の概要(英文)：In this study, we extended a computer language Ruby to increase execution performance to establish a new high-productive computer language HPC Ruby. We successfully developed HPC Ruby static compiler and runtime systems. Based on our evaluation, HPC Ruby showed from 50% to 90% execution performance in numerical programs compared to C language. In order to show feasibility of the use of Ruby in scientific computation, we developed Ruby based software in earth science, astrophysics, and discrete optimization areas.

As for the ultra high-speed distributed computing using 100Gbps internet, we developed 40Gbps single stream TCP system and 100Gbps multi-stream TCP system. We performed very long distance internet experiments using Tokyo, New Orleans (US) and Amsterdam (Europe). Performance results shows we can efficiently utilize more than 90% of available bandwidth in both 40G and 100G environment. This was the world first realization of very long-distance high-speed TCP data-transfer.

研究分野：情報科学, 計算機科学

キーワード：計算機システム 高生産性言語 ネットワーク ハイパフォーマンスコンピューティング 超並列問題  
記述

## 1. 研究開始当初の背景

従来主に用いられてきた、Fortran, C++, Java やそれらの拡張言語は、並列性の活用と、過大な並列性を防ぐ並列性制御が困難であった。スーパーコンピュータが並列化したことに伴い、科学技術プログラムの生産性は却って大きく低下した。科学技術計算の生産性を高めることが急務であるとの認識から、諸外国を中心に並列化コンパイラ開発、並列プログラミングの高生産性を目指した言語開発などが行われた。

しかしながら、このように開発された X10, Fortress, Chapel は、広く現場の研究者・開発者に普及するには多くの問題点があった。(1)利用者層がスーパーコンピュータを使用する計算科学者に限定されるため、ユーザ層の広がり、開発資金、開発層の厚さが限定されること、(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、普及が更に困難となること、(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、スケーラブルなプログラミングをするために必須である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何らの解決策を示さないこと。本研究の目的は、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、広く普及している言語をベースとする高生産性言語を確立することである。本研究では広く普及している汎用言語である Ruby を高速化、数値計算向けに拡張し、上記問題点を根本的に解決する。

## 2. 研究の目的

本研究開発では、関数型オブジェクト指向言語である Ruby を拡張し、スケーラブルである HPC 向け高生産言語として確立する。また、Ruby の特徴である統合的情報環境を生かし、HPC 情報環境における超分散透明性を持つソフトウェア体系を実現する。HPC 向け新言語

の確立と普及には、実際の科学技術問題を Ruby 言語モデルを用いて記述し、並列分散環境で実行することが不可欠である。この目的で、数値シミュレーション等のアプリケーション群を Ruby により問題定式化する。

Ruby 分散実行環境を中心とする成果の実証では、国際連携分散シミュレーション環境等を使用し、超遠距離インターネットを活用した分散処理の通信基盤を構築する。

## 3. 研究の方法

本研究開発では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。高生産性を実現する要件である、

並列・分散コンピューティングにおけるスケーラビリティの実現、厚いユーザ層を持つ言語ベース、コンピュータサイエンスが専門でない科学技術研究者が使いやすいこと、ユーザや、他言語プログラムに対して豊富なインタフェースを持つことは、関数型であり、オブジェクト指向言語である Ruby の特質にマッチする。その成果として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した。

## 4. 研究成果

### ① Ruby 処理系に関する研究

我々は、Ruby 処理系単体の高速化技術、および並列化技術について、(a) 実際的な Ruby 用コンパイラ構成法、および (b) Ruby における並列分散処理環境の構成法について明らかにし、コンパイラを開発した。

(a) 実際的な Ruby 用コンパイラ構成法  
まず、研究を2段階に分けて行った。まずは Ruby 処理系とほぼ完全な互換性を持つ AOT コンパイラの構成法を明らかにした。本研究

では、Ruby 処理系との互換性、及び可搬性に優れた手法を用いて、既存の Ruby プログラムを、Ruby がサポートする全ての環境で高速化することを目標とし開発を進めた。この目標を達成するために、Ruby スクリプトをコンパイルしたバイトコード列を C 言語に変換し、Ruby 処理系の仮想マシン上で動作させる AOT コンパイラを開発した。次に、この研究の知見を生かし、処理系のアドオンとして追加可能な構成方式による、性能解析の結果を用いるコンパイラ Cast0ff を開発した。

(b) Ruby における並列分散処理環境の構成  
本研究課題は、(b-1) MVM による並列化処理系の構成法の研究、および (b-2) 効率的なオブジェクトのプロセス間転送手法の研究に分けて行なった。この成果は、実際に世界中に多くのユーザを抱える Ruby 処理系の上で、実際に利用されることを想定して開発を進めたため、従来の学術研究では得られなかった互換性の維持といった現実的な問題など、多くの新しい問題を明らかにし、我々はそれらに対して適切な手法を提案することができ今後の Ruby 言語に対し大きな広がりを与えた。

#### HPC Ruby に関する研究

Ruby 処理系単体の高速化技術、および並列化技術について、(a) 実際の Ruby 用コンパイラ構成法、および (b) MVM による並列化処理系の構成法について明らかにした。(a) については、3 種のコンパイラを順次開発した。Ruby 処理系とほぼ完全な互換性を持つ AOT コンパイラ。Ruby 処理系との互換性、及び可搬性に優れた手法を用いて、既存の全ての Ruby プログラムを、Ruby がサポートする全ての環境で高速化することを目標とし開発した。この目標を達成するために、Ruby スクリプトをコンパイルしたバイトコード列を C 言語に変換し、Ruby 処理系の仮想マシン上で動作させる AOT コンパイラを開発し

た。次に Ruby の高速化を目的として型解析と静的データフロー析に基づく Ruby の最適化の研究開発を実施した。開発した Ruby 最適化コンパイラの処理概要を図 1 に示す。

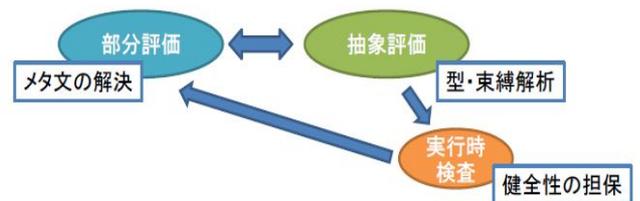


図 1 HPC Ruby 最適化コンパイラの概要  
次いで最適化コンパイラを開発を行うとともに、性能評価を実施した。図 2 に NAS Parallel Benchmarks の結果を示す。図 2 に示されるように、HPC Ruby コンパイラは NPB において、Java よりおおむね高速であり、C の 50% から 80% の性能で実行できることがわかり、Ruby の最大の問題点である性能が著しく低いという問題点が解消されたことが示された。

この成果により、生産性が高く、多くのアプリケーションですでに普及している Ruby 言語に対し、数値計算への適用性を大きく進展させた。

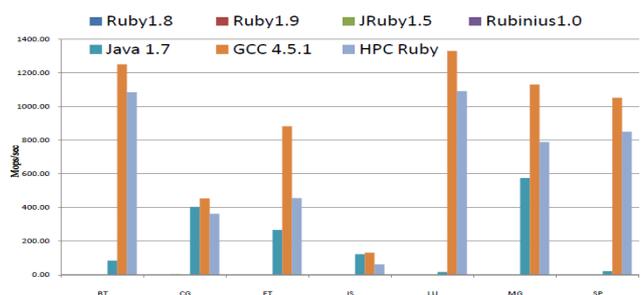


図 2 . 最適化コンパイラ (HPC Ruby) の性能

(1) Ruby を用いた問題定式化に関する研究  
地球惑星科学問題への適用、天体物理学への適用および離散最適化への適用についての研究開発を実施した。

巨大衝突による地球型惑星の衛星形成については、

- 1 . 巨大衝突による地球型惑星の衛星形成月の生成について、地球のマントル起源で

あることをシミュレーション法を開発した。Hosono et al. (2013)は、従来の SPH シミュレーションは、コア・マントルというような密度成層を持つ天体の衝突現象には適さないことを指摘し、密度成層を持つ天体の衝突現象に適した SPH コードを開発した。

また、火星のフォボス、ダイモスは非常に小さな衛星なので、捕獲された衛星という解釈が一般的であったが、われわれは衝突説を検証するために、大粒子 SPH シミュレーションを行った。その結果、ボレアレス盆地から想定される衝突を起こした場合、フォボス、ダイモスを作るのに十分な量の破片が撒き散らされることがわかり、衝突説が強く支持されることになった。

## 2. 太陽系外の大型地球型惑星（スーパーアース）の集積

木星や土星のような巨大ガス惑星は、まずは地球の 10 倍程度の固体惑星が形成され、その強い重力で自らを生んだ原始惑星系円盤の水素・ヘリウムガスを集積して形成される。観測から普遍的に存在することがわかったスーパーアースは地球の 10 倍以上のものも多く、なぜガス惑星にならなかったのが謎であった。われわれは N 体シミュレーションにより、円盤ガスが存在する間は、地球質量程度の原始惑星の軌道が安定に保たれ、円盤ガス消失後にその軌道が不安定化して合体成長してスーパーアースが形成されることを示した（図 3）

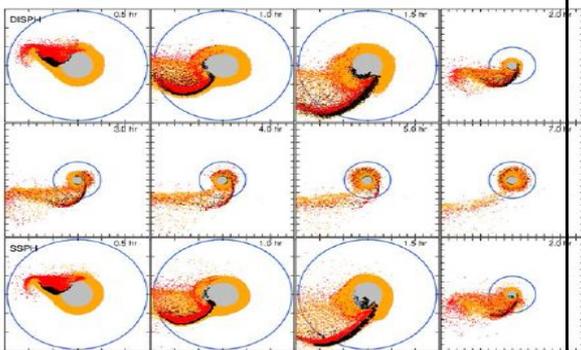


図 3 円盤ガスシミュレーション結果

## 宇宙物理学への適用

本研究では、新しいアルゴリズムにより、移植性が高く、スケーラビリティを改善することを目標とする。これは、従来からそれぞれは広く使われている Barnes-Hut ツリー法と独立時間刻み法を新しいアプローチによって組み合わせるものである。さらに、重力だけでなく、粒子による流体シミュレーションについても、従来から指摘されていた問題点を解決できる新しい方式を研究開発した。まず、移植性、並列スケーラビリティが高く、適用範囲も広い新しい理論天文学シミュレーションアルゴリズムの開発・検討を進めた。このアルゴリズムでは、計算量自体、従来の方法が粒子数の 2 乗で増加していたのを、 $N \log N$  程度にできた(図 4 参照。四角が従来の方法、十字が新しいアルゴリズム)このため、長時間積分でも安定に計算が可能になった。これはまず重力多体系について定式化、テスト実装による評価を進めてきたが、流体を扱う SPH 法についても同様なハミルトニアン分割に基づいた定式化をおこなった。

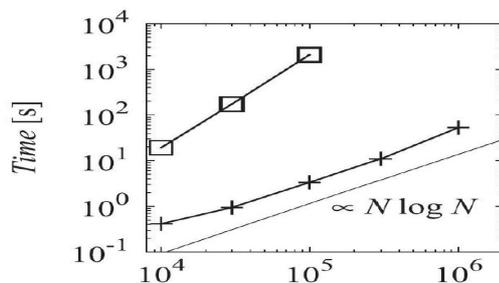


図 4 新アルゴリズムの計算時間

これら 2 分野および離散最適化分野において Ruby の持つ高生産性を活用した問題定式化が実際の問題で有用であることを示し、HPC 分野における Ruby 言語の普及への出発点を与えた。

### (3) 超分散・超並列計算環境による評価

#### 国際分散計算による評価

日米欧を超高速インターネットで結合した評価用システムを構築した。システムは、日本側は国立天文台が運用する Cray-XT6 型スーパーコンピュータを用い、オランダ側はア

ムステルダム大学が運用するホイヘンス・スーパーコンピュータ（IBM 製）を用い、両者を結合する超遠距離インターネット通信には本プロジェクトで開発した超高速・超遠距離通信ファシリティを用いた。評価は、アムステルダム大学の S.F.Portegies Zwart 教授と連携して、2 台のスーパーコンピュータを協調連携させ、天文学シミュレーションを実施した。本実験の成果として、私たちが確立した超遠距離・超高速 TCP 通信技術を用いることで日本・オランダという大きな距離を隔てたスーパーコンピュータを協調連携させることが可能となり、1 台では不可能な大規模シミュレーションが実施可能となることが示された。

超遠距離データ通信に関する研究開発超遠距離・超高速データ通信ソフトウェアの開発を行い、実証実験として、SC14 国際会議展示場とアムステルダム市間で実施した国際実証実験を実施した。

まず、TCP 関連ソフトウェアとして、ソフトウェアによる動的ペーシングを開発し Linux に実装した。動的ペーシングはネットワークの状況に応じてパケット間隔を調整する。具体的には、輻輳ウィンドウサイズと往復遅延を用いる。本来 TCP は、輻輳ウィンドウサイズ分のパケットを一度に送信して、往復遅延分の時間待ち、再び輻輳ウィンドウサイズ分のパケットを送信する、という処理を繰り返す。

実証実験として SC14 において大陸間通信実験を実施した。本実験では静的ペーシングに加えて、動的にパケット間隔を調整するソフトウェアによる動的ペーシングの評価も行った。静的ペーシングが NIC の機能を用いているのに対し、動的ペーシングはソフトウェア実装されているため、ハードウェアに依存せず広く用いることができる。実験では、ネットワークの端点はそれぞれ SC14 の会場であるニューオーリンズと、アムステルダムに

設置されており、それらをつなぐネットワークの往復遅延はおよそ 110ms であった。

PC は 40Gbps のネットワークを介して各スイッチに接続されており、スイッチ同士は 100Gbps 回線で接続されている。図 5 は二つのネットワークをつなぐ回線の概略を示している。ニューオーリンズから出た回線はニューヨークに接続されており、ニューヨークから出た回線は大西洋を経てアムステルダムに到着する。



図 5. SC14 国際実験で用いた経路

TCP は回線に対して 40%程度の性能しか得られていない一方で、動的ペーシングを用いた場合、回線 85%程度の性能が得られた。また静的ペーシングを行った場合、回線の 97%の性能が得られた。ペーシングを用いない場合に性能が低下してしまうのは TCP のバースト性が原因だと考えられる。動的ペーシングはバースト性を除去することによって性能の低下を防止している。

## 5. 主な発表論文等

雑誌論文（合計 48 編）

1. 芝哲史, 笹田耕一, 平木敬:CastOff: Ruby 用コンパイラのライブラリとしての実装, 情報処理学会論文誌 (PRO) Vol.5 No.3 pp.1-22 (2012.8)
2. 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 (2012.3).
3. 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 (2011.3).

4. Hosono, N., Saitoh, T. R., and Makino, J., "Density-Independent Smoothed Particle Hydrodynamics for a Non-Ideal Equation of State", PASJ, 65, 108 arXiv:1307.0916 [astro-ph.IM] (2013)
5. Saitoh, T. R. and Makino, J., "A Density-independent Formulation of Smoothed Particle Hydrodynamics", ApJ, 768, 44, (2013)
6. Citron, Robert I.; Genda, Hidenori; Ida, Shigeru: Formation of Phobos and Deimos via a giant impact Icarus, Volume 252, p. 334-338 (2015) 10.1016/j.icarus.2015.02.011
7. Lewis, K. M.; Ochiai, H.; Nagasawa, M.; Ida, S. Extrasolar Binary Planets II: Detectability by Transit Observations, The Astrophysical Journal, Volume 805, Issue 1, article id. 27, 13 pp. (2015) 10.1088/0004-637X/805/1/27
8. Matsumura, Soko; Ida, Shigeru; Nagasawa, Makiko Effects of Dynamical Evolution of Giant Planets on Survival of Terrestrial Planets. The Astrophysical Journal, Volume 767, Issue 2, article id. 129, 14 pp. (2013) 10.1088/0004-637X/767/2/129
9. Jesper Jansson, Kunihiko Sadakane, Wing-Kin Sung. "Ultra-succinct representation of ordered trees with applications", Journal of Computer and System Sciences, 78(2):619-631, 2012.
10. Tomohiro Sonobe and Mary Inaba, "Division and Alternation of Decision Variables", Pragmatics of SAT 2012(2012).
11. Tomoaki Ishiyama, Junichiro Makino, Simon Portegies Zwart, Derek Groen, Keigo Nitadori, Steven Rieder, Cees de Laat, Stephen McMillan, Kei Hiraki, Stefan Harfst, "The Cosmogrind Simulation: Statistical Properties of Small Dark Matter Halos" The Astrophysical Journal, DOI: 10.1088/0004-637X/767/2/146. (2011)
12. 森藤福真, 平木敬, "TCP 通信における機械学習を用いたロス分類アルゴリズム" 信学技報, CPSY2013-19, pp55-60, (2013).  
学会発表(56件)および図書(0件) スペースの都合により省略。  
〔図書〕(計0件)  
〔産業財産権〕  
出願状況(計0件) 取得状況(計0件)  
〔その他〕  
報道関連情報、アウトリサーチ活動情報  
1. Ruby 処理系をパブリックドメインソフトとして公開 (笹田耕一)
6. 研究組織  
(1) 研究代表者  
平木 敬 (HIRAKI KEI), 東京大学・情報理工学系研究科、教授、研究者番号: 20238348  
(2) 研究分担者  
笹田 耕一 (SASADA KOICHI), Heroku Inc.・Engineer, 研究者番号 10436561 (当初研究分担者、所属変更により研究協力者に変更)  
  
定兼 邦彦 (SADAKANE KUNIHICO), 東京大学・情報理工学系研究科、教授、研究者番号: 20323090  
  
牧野 淳一郎 (MAKINO JUNICHIRO), 理化学研究所・計算科学研究機構、副プロジェクトリーダー、研究者番号: 50229340  
  
井田 茂 (IDA SHIGERU), 東京工業大学・理工学研究科、教授、研究者番号: 60211736  
稲葉 真理 (INABA MARY), 東京大学・情報理工学系研究科・准教授、研究者番号: 60282711