

オブジェクト指向スクリプト言語 Ruby の処理系の刷新

YARV: Yet Another RubyVM

1. 背景

プログラミング言語 Ruby はまつもとゆきひろ氏によって設計、開発された日本発のオブジェクト指向スクリプト言語である。その処理系は日本発のオープンソースソフトウェアであり、Ruby の言語としての秀逸さより、世界中で広く利用されている。とくに最近では Ruby on Rails というウェブアプリケーションフレームワークの爆発的な人気によって、Ruby も更に注目を浴びている。しかし、現在の Ruby 処理系は実行速度が十分でないという問題があった。この問題を解決するために昨年度、2004 年度未踏ソフトウェア創造事業未踏コースにおいて、「Ruby プログラムを高速に実行するための処理系の開発」というプロジェクトで Ruby プログラムを高速に実行する処理系 YARV: Yet Another RubyVM を開発した。

しかし、開発した YARV は既存の Ruby 処理系の拡張機能として実現していたため、(a) 気軽にインストールできない (b) スレッドやリフレクションなど、Ruby のコア機能を実現できない、などの問題がある。つまり、Ruby 処理系として十分なものではなかった。

2. 目的

本プロジェクトの最終目的は、現在ある YARV の問題点を解決し、次期 Ruby 公式実装としてリリースすることである。

そこで、本年度は以下の項目を実装することとした。まず、Ruby 処理系としての完成度を高め、YARV を統合した機能的に不足のない処理系を実装する。その際必要になるスレッドの実装では従来のユーザレベルスレッドではなく、OS がサポートするネイティブスレッドを用いて行い、OS が提供する機能を利用できるようにする。さらに、現在の処理系の機能拡張として、複数のインタプリタ、つまり VM を一つのプロセス内で扱うための Multi-VM インスタンスに対応する。

3. 開発の内容

本開発では、高速化よりも「使い物にする」ことを念頭に、次に示す開発目標に取り組んだ。

具体的な目標としては、最優先事項として Ruby 本体に YARV 処理系を融合し刷新する。これにより、現行 Ruby の速度的な問題点を克服する一方、YARV の Ruby 処理系として不十分であった点を解決する。Ruby 処理系としての完成度については、網羅的な自動テストを作成し、検証する。このテストにパスすることを最優先とする。引き続き、OS のネイティブスレッド対応、Multi-VM インスタンス機能の実装を行い、Ruby 処理系としての互換性向上や機能強化を図る。具体的な数値目標としては、2 プロセッサ構成の計算機を用いた

ネイティブスレッド対応 Ruby により従来比 1.5 倍以上の性能向上を目指す。Multi-VM インスタンス機能については Apache 用拡張モジュールである mod_ruby に対応させ、アプリケーションごとに独立した VM を割り当てることを目標とする。以上を目標として設定した。

成果としては、最優先事項の YARV の Ruby 処理系への統合はほぼ完了した。完成度を確認するために用意した約 2000 の自動テストも 99%以上パスしている。

ネイティブスレッドは POSIX Thread (UNIX 系)、Win32 Thread (Windows 系)を用いる実装を行った(図 1)。ただし、Ruby レベルの並列実行には対応しないグローバル VM ロックを用いた実装とした。しかし、拡張ライブラリによる並列実行をサポートするための API を提供し、これを用いた拡張ライブラリの実験の結果 Dual Core CPU 上で約 2 倍の性能向上を確認した。

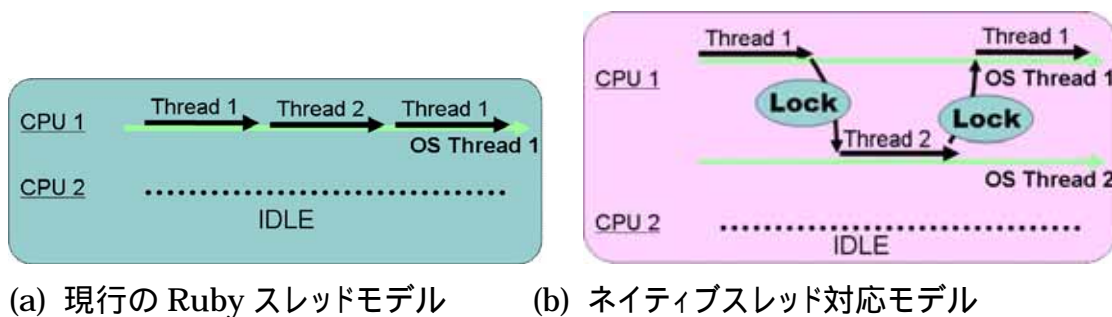


図1 ネイティブスレッド対応

Multi-VM インスタンス機能については暫定的に対応したが、目標としていた mod_ruby の対応までには至らなかった。各 VM の並列実行をサポートし、その結果約 1.5 倍の性能向上達成を確認した。

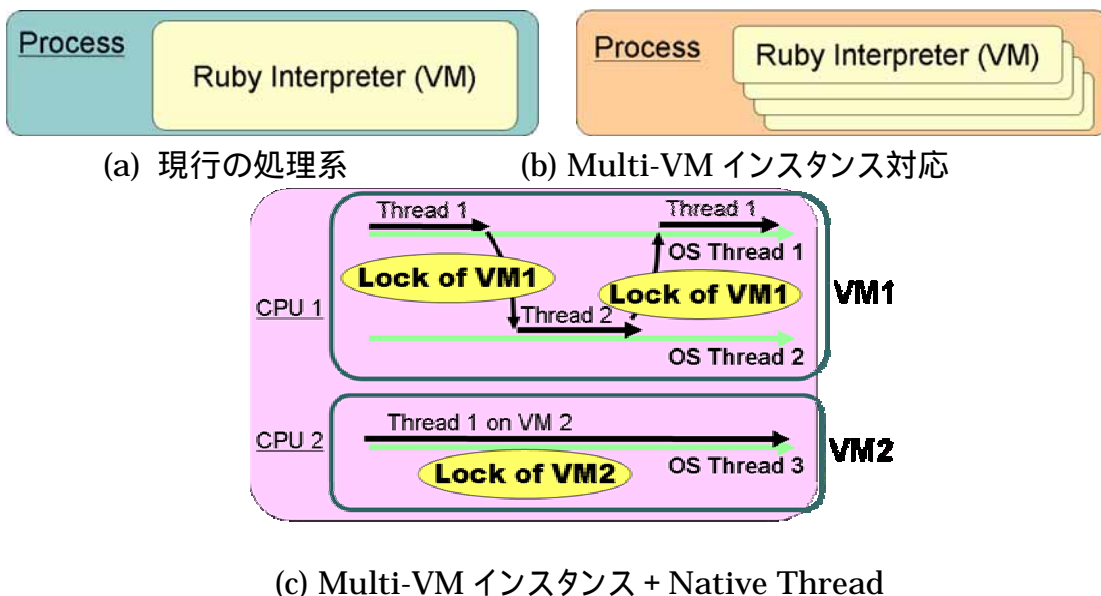


図2 Multi-VM インスタンスのサポート

今後は、すでに本開発の成果が取り込まれることが決定している次期公式リリースへ向けて、残っているいくつかの課題を着実に解決していく。

4. 従来技術(または機能)との相違

本開発の最大の成果は、本開発の目的である YARV を現在の Ruby 処理系と統合し、Ruby 処理系としての完成度を大幅に向上したことである。本開発によって昨年度の限定的な速度の改善を更に推し進め、処理速度が遅いという Ruby の大きな欠点を改善できた。これで従来言われてきた「Ruby は実行速度が遅いので使えない」というレッテルを返上することになる。

5. 期待される効果

本開発による Ruby 処理系の改善は、国際的に利用されている Ruby がさらに競争力を増すことを意味する。たとえば、Ruby の比較対象として挙げられる Perl や Python などの言語の処理系に対しても、プログラムの実行速度は同等かそれ以上のものとなり、十分な競争力を得たと言える。

ネイティブスレッド対応は、これから主流になるだろう並列計算機を有効に使うために必須の機能である。現状では Ruby レベルでの並列実行には対応しないが、それを対応するための基盤は十分築くことができた。今後、Ruby プログラムの並列実行に向けて作業を進めることが可能になる。

Multi-VM インスタンスの十分な対応が済めば、Ruby 処理系のアプリケーションへ組み込みが容易となり、Ruby の用途が大きく広がることになる。

6. 普及(または活用)の見通し

Ruby 2.0 もしくは 1.9 の公式リリースにおいて本処理系の開発成果が取り込まれることが決定している。これが実現した際には世界中の Ruby ユーザが本処理系を利用することになる。本処理系は他の競合するようなスクリプト言語と比較しても十分高速であり、Ruby という言語の魅力がより向上する。その結果、日本発のオープンソースソフトウェア Ruby の評価が、世界的にもますます高くなる。

7. 開発者名(所属)

笹田耕一(東京農工大学大学院)

(参考)

YARV: Yet Another RubyVM <http://www.atdot.net/yarv/>