

Raku のサーバーを使った実行

175748C 氏名: 大蔵 海斗 指導教員: 河野 真治

2020/09

1 Raku の高速実行

現在開発の進んでいる言語にスクリプト言語の Raku がある。Raku は起動時間が Perl5 や Python, Ruby などの主要なスクリプト言語に比べて非常に低速である。本研究では、この問題を解決するために、予め Raku のコンパイラを立ち上げたサーバーを立ち上げ、そのサーバー上でコンパイルを行う手法の提案を行う。

2 Raku

Raku は元は Perl5 の後継言語の Perl6 として開発されていたが、言語仕様及び処理実装が Perl5 と大幅に異なっており、言語的な互換性が存在しない。従って現在では Raku と Perl5 は別言語としての開発方針になっている。Raku という名称は、現在有力な処理系である Rakudo が由来となっている。Raku の言語的な特徴としては、独自に Raku の文法の拡張が可能な Grammar, Perl5 と比較した場合のオブジェクト指向言語としての進化も見られる。また Perl6 は漸進的型付け言語である。従来の Perl の様に変数に代入する対象の型や、文脈に応じて型を変更する動的型言語としての側面を持ちつつ、独自に定義した型を始めとする様々な型に、静的に変数の型を設定する事が可能である。Raku の起動は、MoarVM を起動、nqp をロード、Rakudo をロードもしくはコンパイルし、その後 JIT しながら実行する。

3 Rakudo

Rakudo とは Raku の現在の主流な実装である。(Raku は言語名, Rakudo はコンパイラ) Raku は

仕様と実装が明確に区分されており、Rakudo という実装、roast という Raku の仕様 (テストスイートがある)。Rakudo の構成は、MoarVM と呼ばれる Rakudo のために構築された VM, NQP と呼ばれる Raku のサブセット, NQP と Raku 地震で記述された Raku である。Rakudo は MoarVM の他に JVM や JavaScript を動作環境として選択可能である。

4 MoarVM

MoarVM は Rakudo, NQP のために構築された VM である。C 言語で実装されている。JIT コンパイルなどが現在導入されているが、起動時間が低速であるなどの問題がある。MoarVM 独自の ByteCode があり、NQP からこれを出力する機能などが存在している。MoarVM は NQP と ByteCode を解釈する。

5 NQP

NQP とは Not Quite Perl の略で Raku のサブセットである。その為基本的な文法などは Raku に準拠しているが、変数を束縛で宣言するなどの違いが見られる。この NQP で記述された Raku の事を Rakudo と呼ぶ。Rakudo における NQP は現在 MoarVM, JVM 上で動作する。NQP は最終的には NQP 自身でブートストラップする言語であるが、ビルドの最初にはすでに書かれた MoarVM のバイトコードを必要とする。Raku の一部は NQP を拡張したもので書かれている為、Rakudo を動作させる為には MoarVM などの VM, VM に対応させる様にビルドした NQP がそれぞれ必要となる。

6 Raku が遅い原因

通常, Ruby のようなスクリプト言語ではまず VM が起動し, その後スクリプトをバイトコードに変換して実行という手順を踏む. 対して Raku は, コンパイラの Rakudo 自体が Raku と NQP で書かれているため, MoarVM を起動し, Rakudo と NQP のバイトコードを読み取り, Rakudo を起動し, その後スクリプトをバイトコードに変換して実行という手順を踏む. そのため, Rakudo はインタプリタの起動時間及び, 全体的な処理時間が他のスクリプト言語と比較して非常に低速である. また, Raku は実行時の情報が必要であり, メソッドを実行する際に `invoke` が走ることも遅い原因である. `invoke` は MoarVM の `method` 呼び出しのバイトコードである.

7 Abyss サーバー

Abyss サーバーはクライアント側から投げられた Raku スクリプトを実行するためのサーバーである. Abyss サーバーはユーザーが Raku を直接立ち上げるのではなく, まず同一ホスト内で Abyss サーバーを起動し, ユーザーは Abyss サーバーにファイルパスをソケット通信で送り, Abyss サーバーがファイルを開き実行し, その実行結果をユーザーに返す. この手法を用いることで, サーバー上で事前に起動した Rakudo を再利用し, 投げられた Raku スクリプトの実行を行うため, Rakudo の起動時間を短縮できると推測できる.

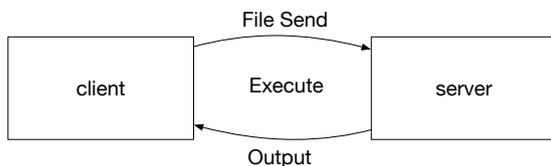


図1 Abyss サーバーを用いた Raku の実行

8 Abyss サーバーの課題