

# Raku(Perl6) のサーバーを使った実行

## Running Raku(Perl6) using Raku server

165727F 氏名: 福田 光希 指導教員: 河野 真治

2019/10

### 1 研究目的

Perl5 の後継言語として, Raku が開発されている. Raku は MoarVM という VM 上で動作するが, 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遅い. その為, 本研究では Raku プログラムの新たな実行手法の提案, また OS 上でスクリプト言語を実行する最適な方法の見直し, スクリプト言語をサーバーで実行する利点と欠点の考察を行う.

### 2 Raku

Perl の現在の主流な実装は Rakudo である. Rakudo は MoarVM, と NQP と呼ばれる Perl6 のサブセット, NQP で記述された Perl6 という構成である. MoarVM は NQP を解釈する. この NQP で記述された Perl6 の事を Rakudo と呼ぶ. Rakudo は MoarVM の他に JVM, Javascript を動作環境として選択可能である. 言語的な特徴では Perl5 とは違いオブジェクト指向のサポートが強力になり, 漸進的型付け言語としての特徴を持つ.

現状の Perl6 の実行は Perl6 で記述されたコンパイラを load して JIT しながら実行すること自体に時間がかかっている.

### 3 MoarVM

MoarVM は Perl6 に特化した 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, 起動時間などが低速である問題がある. 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.

### 4 Perl6 サーバーの実行

### 5 まとめ

中間予稿までに Perl6 スクリプトを投げて実行するサーバーの実装, および「自分でプロセス立ち上げて Perl6 実行

する手法」と「既にあるサーバーに投げて Perl6 スクリプトを実行する手法」の差を測るために時間の計測を行った.

今回実装したサーバーでは, 別のスクリプトを実行する前にサーバーの状態を初期状態にリフレッシュできていないので, 信頼性が保てていない, この問題を解決するためにサーバーに投げられた Perl6 をコンパイラで実行する際のセパレーションを保証する機構を作ることが課題としてあげられる. また今回実装したサーバーとの通信は TCP ソケット通信であり, 現状の Perl6 には Unix domain ソケット通信がないので Unix domain ソケットを実装する予定である. 加えて Unix domain socket の permission を自分だけが実行できるように書き換える, 外部から不正なアクセスを行わせないことも必要である.

### 参考文献

- [1] Andrew Shitov. Perl6 Deep Dive
- [2] 清水隆博, 河野真治. CbC を用いた Perl6 処理系. 琉球大学工学部情報工学科平成 30 年度学位論文 (学士), 2018.