

# Rubyアソシエーション開発助成2019 中間報告

## 題目: 複数ファイルに対応したプリコンパイル機構の作成

報告者: 永山 涼雅

### 報告概要

既存のMRIを拡張することで複数のInstructionSequence (ISeq) を単一のバイナリに出力する機能を実装した。また、その機能を利用し、アプリケーションなどに含まれるRubyスクリプトファイルを単一のバイナリファイルにまとめキャッシュすることで、起動の高速化を図るライブラリを作成した。

Railsアプリケーションを用いた実験では、Bootsnapと比較してコンパイルキャッシュのサイズは12.5%程度小さくなったが、一方で起動時間がBootsnapを用いたものに比べ1.6倍以上伸びることを確認した。

速度低下の原因はバイナリファイル内の局所性が低下することでメモリのキャッシュミスが増加したためだと考えられる。これを解消するのは困難なため、プロジェクトを中止する。

## 1. プロジェクト概要

RubyVMの中間表現であるISeqはバイナリ表現と相互に変換することができ、Rails 5.2以降でプロジェクトにデフォルトインストールされるBootsnap<sup>[1]</sup>はそのバイナリ表現をキャッシュしておくことでアプリケーションの起動速度を改善している。しかし、Bootsnapが行っているのは.rbファイルごとのキャッシュであり、gemやプロジェクトといった複数ファイルからなる単位でキャッシュを行うことはできない。また、Rubyインタプリタが起動時にロードする一部のgemはBootsnapの初期化前に読み込まれるため仕組み上キャッシュされない。

本プロジェクトでは、MRIの機能を拡張することで複数のスクリプトファイルからなるgemやプロジェクトを単一のバイナリファイルにプリコンパイルするための機構を提供し、Rubyアプリケーションの起動の高速化とキャッシュサイズの削減を目的とする。

## 2. 実装

ISeqとそのバイナリ表現の変換に関するAPI、`RubyVM::InstructionSequence#to_binary`および`RubyVM::InstructionSequence.load_from_binary`は内部でそれぞれ、`ibf_dump`、`ibf_load` という構造体を利用してシリアライズ・デシリアライズを行っている。

本プロジェクトでは、これらの構造体を利用して複数のISeqから単一のバイナリへの変換、および、バイナリからISeqへの逆変換を行うAPIをRubyスクリプトに提供するパッチを作成した<sup>[2]</sup>。

また、それらのAPIを用いてBootsnapと同じようにコンパイルキャッシュを作成することで、アプリケーションの起動を高速化する簡易的なライブラリを作成した<sup>[3]</sup>。

## 3. 評価

rails new コマンドで作成した空のRailsプロジェクトを用いて起動時間を評価する。

### 環境

- OS: macOS Catalina
- CPU: Intel Core i7
- メモリ: 16GB
- Rails: 6.0.1

Bootsnapを用いた場合の起動時間が平均1.389秒なのに対して、作成したライブラリを用いた場合の起動時間は平均2.278秒となった。このときのキャッシュミスの回数はそれぞれ2,895,329回、4,711,601回であった。

また、生成されたコンパイルキャッシュのサイズはそれぞれ 6.9 MB、7.9 MBであった。

下記に示す変更などを行ったが、大きな改善はみられなかった。

### 3.1. 試みた変更

#### 3.1.1. mmap() の利用

ファイルの読み込みを一括で行うのではなく、mmapを用いてバイナリファイルをメモリ上にマッピングすることで逐次的に行うように変更した。

#### 3.1.2. Lazy loading の利用

MRIにはISeqのデシリアライズをISeqが実際に利用されるまで遅延させるLazy loadingという機能があり、コンパイル時に特定のオプションを与えることでこれを有効化できる。これを利用するように変更した。

#### 3.1.3. 出力されるバイナリの構造を変更

現在のISeqのバイナリ構造では、ISeq自体とISeq内で利用されるオブジェクトのバイナリ表現は離れた位置に出力される。オブジェクトの出力位置を工夫することでISeqとオブジェクトが近接した位置に出力されるように変更した。

## 4. 結論

複数のISeqを単一のバイナリに出力する機能を作成した。

この機能を利用したところ、空Railsプロジェクトで約12.5%コンパイルキャッシュのサイズを削減することができた。一方で、起動時間は1.64倍長くなった。

起動速度低下の原因は局所性が低下したことによるキャッシュミスの増加であると考えられる。

これを解消することは困難なため、本プロジェクトを中止する。

## 参考

[1]: <https://github.com/Shopify/bootsnap>

[2]: <https://github.com/NagayamaRyoga/ruby/tree/dumper>

[3]: <https://github.com/NagayamaRyoga/nidone>