

# Goodbye fat gem

Sutou Kouhei

*ClearCode Inc.*

*RubyKaigi Takeout 2020*  
*2020-09-04*

# Sutou Kouhei

The president of ClearCode Inc.

クリアコードの社長

Silver Sponsors



ClearCode Inc.

<https://www.clear-code.com/>

Free software is important in ClearCode. We develop/support software with our free software development experiences. We feed back our business experiences to free software.

# Sutou Kouhei

- ✓ The maintainer of rake-compiler  
rake-compilerのメンテナー
- ✓ A gem for generating fat gem  
fat gemを作るためのgem
- ✓ The maintainer of Ruby-GNOME  
Ruby-GNOMEのメンテナー
- ✓ A project that used fat gem  
fat gemを使っていたプロジェクト

# Fat gem

Gem that includes  
pre-built binaries

ビルド済みバイナリー入りのgem

# Why is fat gem needed?

どうしてfat gemが必要なのか

- ✓ Difficult to build extension library  
拡張ライブラリーのビルドが難しい
- ✓ Extension library:  
A Ruby library implemented with C API  
拡張ライブラリー：C APIを使って実装されたRubyライブラリー
- ✓ Fat gem just copies pre-built binaries  
fat gemは単にビルド済みバイナリーをコピーするだけ（ビルドしない）

# Why is building it difficult?

どうして拡張ライブラリーのビルドは難しいのか

- ✓ Users need build environment  
ユーザーはビルド環境を用意しないといけない
  - ✓ e.g.: C compiler, make and so on  
例: Cコンパイラーやmakeなど
- ✓ Users need build dependencies  
ユーザーが依存するライブラリーを用意しないといけない
  - ✓ e.g.: GTK+ 3 for gtk3 gem  
例: gtk3 gemはGTK+ 3が必要

# With fat gem

## fat gemを使うと

- ✓ Users don't need build environment!  
ユーザーはビルド環境を用意しなくてもよい
- ✓ No C compiler  
ユーザーはビルド環境がなくてもよい
- ✓ Users don't fail installation!  
ユーザーはインストールに失敗しない
- ✓ Fat gem just copies pre-built binaries  
fat gemは単にビルド済みバイナリーをコピーするだけ

With fat gem  
fat gemを使うと

Happy!



...Really?  
…ほんとに？



# A fat gem maintainer says...

あるfat gemメンテナー曰く...

# Thanks fat gem! Goodbye fat gem!

ありがとうfat gem!  
さよならfat gem!

# Fat gem problem 1

fat gemの問題1

Can't use  
the latest Ruby  
immediately

いち早く最新のRubyを使えない

# Details

## 詳細

- ✓ Ruby is released every Christmas  
Rubyは毎年クリスマスにリリースされる
- ✓ To use fat gems with the latest Ruby:  
最新のRubyでfat gemを使うには
  - ✓ Need to release fat gems for it  
最新のRuby用のfat gemがリリースされていないといけない
- ✓ Users can't use the latest Ruby while:  
ユーザーは↓の間は最新のRubyを使えないまま
  - ✓ Any of fat gems doesn't support it  
使っているfat gemのどれか1つでも最新のRubyをサポートしていない

# From fat gem maintainer view

fat gemメンテナー視点

Not all fat gem  
maintainers  
can do it  
**immediately**

すべてのfat gemメンテナーが**すぐ**に対応できるわけではない

# Fat gem problem 2

fat gemの問題2

Vulnerability  
response  
may get delayed

脆弱性対応が遅れがち

# Details

## 詳細

- ✓ Only bindings fat gem case  
バインディングのfat gemだけのケース
  - ✓ Bindings: A extension library to use an external library  
バインディング：外部のライブラリーを使う拡張ライブラリー
- ✓ Includes the external library binary  
fat gem内に外部ライブラリーのバイナリーも含む
- ✓ When a vulnerability of the external library is found:  
外部ライブラリーに脆弱性が見つかった場合：
  - ✓ Should be released immediately with fix  
すぐに修正を含んだバージョンをリリースすべき

# From fat gem maintainer view

fat gemメンテナー視点

Not all fat gem  
maintainers  
can do it  
**immediately**

すべてのfat gemメンテナーが**すぐ**に対応できるわけではない

# Fat gem problem 3

fat gemの問題3

Can't control  
the external  
library version

外部ライブラリーのバージョンをコントロールできない



# Details

## 詳細

- ✓ Only bindings fat gem case  
バインディングのfat gemだけのケース
- ✓ Maintainers choose one external library version  
メンテナーが外部ライブラリーのどのバージョンを使うかを選ぶ
- ✓ If old external library is chosen, users can't use the latest version  
古い外部ライブラリーを選んだら、ユーザーは最新バージョンを使えない

# Fat gem problem 4

fat gemの問題4

require is slower

requireが遅くなる

# require for fat gem

fat gem用のrequire

Fat gem needs the following code:

fat gemでは次のようなコードが必要:

```
begin
  # Try the bundled binary in fat gem
  require "#{RUBY_VERSION[/\d+\.\d+/]}/io/console.so"
rescue LoadError
  # Use the local built binary
  require 'io/console.so'
end
```

# Why slower?

## なぜ遅くなるか

- ✓ For not fat gem install:  
fat gemを使わない環境：
  - ✓ Such as on GNU/Linux and macOS  
たとえばGNU/LinuxやmacOS
  - ✓ require for fat gem is always failed  
fat gem用のrequireは必ず失敗
- ✓ Can't ignore with large \$LOAD\_PATH:  
\$LOAD\_PATHが大きい場合は無視できない
  - ✓ e.g.: +0.1s with Ruby on Rails application  
例：Ruby on Railsアプリケーションでは0.1秒遅くなる  
<https://github.com/ruby/io-console/pull/4>

# Fat gem problem 5

fat gemの問題5

Fat gem release  
may be forgotten

fat gemのリリースを忘れられる

# Details

## 詳細

- ✓ Normal release is easy  
通常のリリースは簡単
  - ✓ rake release is done in a few seconds  
rake releaseは数秒で終わる
- ✓ Fat gem release isn't easy  
fat gemのリリースは簡単じゃない
  - ✓ rake-compiler-dock help maintainers but...  
rake-compiler-dockを使えばマシになるけど...
  - ✓ It will take at least a few minutes...  
少なくとも数分かかる...

# From fat gem maintainer view

## fat gemメンテナー視点

- ✓ Don't want to release a new version...  
新しいバージョンをリリースしたくないな...
- ✓ Ruby-GNOME case: 10+ related gems  
Ruby-GNOMEの場合：10以上の関連gemがある
- ✓ It takes at least 30 minutes  
順調にいった場合でも少なくとも30分はかかる
- ✓ Forget to release a fat gem  
fat gemをリリースするのを忘れる
- ✓ io-console 0.4.8 case  
<https://github.com/ruby/bigdecimal/pull/148#issuecomment-512075494>

# Fat gem problem 6

fat gemの問題6

High  
maintenance  
cost

メンテナンスが大変



# Details

## 詳細

- ✓ Especially binding fat gem case  
特にバインディングのfat gemのケース
- ✓ Normally, cross-compiling is used  
通常、クロスコンパイルしてfat gemを作る
- ✓ Most external libraries don't do it  
多くの外部ライブラリーはクロスコンパイルなんてしない
- ✓ There are some problems on upgrading  
外部ライブラリーのバージョンをあげるたひになにかしら問題が見つかる
- ✓ Ruby-GNOME has 10+ related external libraries  
Ruby-GNOMEは10以上の関連外部ライブラリーがあって大変だった

# How to solve these problems?

これらの問題を解決するには？

Thanks fat gem!  
Goodbye fat gem!

ありがとうfat gem!  
さよならfat gem!

# Why is fat gem needed

fat gemが必要だった理由

- ✓ Users don't have build environment  
ユーザーがビルド環境を持っていない
- ✓ Especially, Windows users  
特にWindowsユーザー

# Windows users and build env

## Windowsユーザーとビルド環境

- ✓ RubyInstaller for Windows provides build env by default since 2.4  
RubyInstaller for Windows 2.4以降はデフォルトでビルド環境を提供
- ✓ Ruby 2.3 reached EOL  
Ruby 2.3はEOLになっている
- ✓ Most Windows users must use 2.4 or later  
I know some Windows users don't use RubyInstaller for Windows  
ほとんどのWindowsユーザーは2.4以降を使っているはず
- ✓ Most Ruby users have build env now!  
ほとんどのユーザーはビルド環境を持っている！

# Ruby-GNOME said goodbye fat gem!

Ruby-GNOMEはfat gemにさようならをした!

✓ Since 2018-10-31  
2018-10-31から

✓ Install related issues isn't increased  
インストール関連のissueは増えていない

# Resolve fat gem problem 1

fat gem関連の問題の解決1

Can use  
the latest Ruby  
immediately

いち早く最新のRubyを使える

# Details

## 詳細

- ✓ Don't need a new release for new Ruby  
新しいRuby用にgemをリリースする必要がない
- ✓ Until new Ruby doesn't change C API  
新しいRubyでC APIが変わっていない限り
- ✓ Can release a new version before new Ruby release  
新しいRubyがリリースされる前に新しいバージョンのgemをリリースできる
- ✓ Can test with preview release Ruby on CI  
CIでpreviewリリースのRubyをテストできる

# Resolve fat gem problem 2

fat gem関連の問題の解決2

Vulnerability  
response  
can be done by  
each system

脆弱性は各システムが対応してくれる



# Details

## 詳細

- ✓ Especially, binding fat gem case  
特にバインディングfat gemの場合
- ✓ Packaging system will update soon than binding fat gem maintainers  
バインディングfat gemのメンテナーよりもパッケージングシステムの方がすぐに更新することが多い
- ✓ The # of packaging system maintainers is larger than the # of maintainers of each fat gem  
各fat gemのメンテナーの数よりもパッケージングシステムのメンテナーの数の方が多いため

# Resolve fat gem problem 3

fat gem関連の問題の解決3

Can control  
the external  
library version

外部ライブラリーのバージョンをコントロールできる

# Details

## 詳細

- ✓ Packaging system may provide the latest external library  
パッケージングシステムは最新の外部ライブラリーを提供してくれるかもしれない
- ✓ Users can build suitable version of the external library  
ユーザーは適切なバージョンの外部ライブラリーをビルドできる

# Resolve fat gem problem 4

fat gem関連の問題の解決4

require isn't slower

requireが遅くならない

# Details

## 詳細

No fallback require:  
フォールバック用のrequireがいない

```
# Always use the local built binary  
require 'io/console.so'
```

# bigdecimal said goodbye fat gem!

bigdecimalはfat gemにさようならをした!

- ✓ Because of this  
これが理由
- ✓ [ruby/bigdecimal#149](#)

# Resolve fat gem problem 5, 6

fat gem関連の問題の解決5, 6

# Easy to maintain

メンテナンスしやすくなる

# Details

## 詳細

- ✓ **No fat gem release process**  
fat gemをリリースする作業がなくなる
- ✓ **Don't forget fat gem release 😊**  
fat gemのリリースも忘れない 😊
- ✓ **No cross compiling**  
クロスコンパイルしなくてもよい
- ✓ **No rake-compiler**  
rake-compilerもいらない
- ✓ **Maintainers can use their time for others**  
メンテナは別のことに時間を使える



# Stop fat gem

fat gemをやめると

- ✓ All problems can be solved!  
すべての問題を解決できる！
- ✓ Additional good points  
さらにもいいことも
  - ✓ Enable optimization for each user's env  
各ユーザーの環境ごとに最適化できる
  - ✓ e.g.: `-O3 -march=native` GCC options  
例：GCCの`-O3 -march=native`オプションを使う

Stop fat gem  
fat gemをやめると

Happy!



...Really?  
...ほんとは？

# Stop fat gem problem 1

fat gemをやめたときの問題1

## Long install time

インストール時間が長くなる

# Details

## 詳細

### ✓ Fat gem:

- ✓ Just copy pre-built binaries  
単にビルド済みバイナリーをコピー

### ✓ Gem:

- ✓ Build and install binaries  
バイナリーをビルドしてインストール

# How to resolve this?

解決法は？

# No idea...

なんかある？

# Stop fat gem problem 2

fat gemをやめたときの問題2

# Fail to install

インストールに失敗する

# Details

## 詳細

- ✓ Only bindings gem case  
バインディングのgemだけのケース
- ✓ Needs the external library to build  
ビルドするために外部ライブラリーが必要
  - ✓ If it doesn't exist, gem install is failed  
外部ライブラリーがないとgem installが失敗
  - ✓ e.g.: `gem install rmagick` is failed without ImageMagick  
例: ImageMagickがないと`gem install rmagick`が失敗

# How to resolve this?

解決法は？

# Install the external library automatically

自動で外部ライブラリーをインストール



# RubyInstaller for Windows

Put the external library package name to the gem's metadata:

gemのメタデータに外部ライブラリーのパッケージ名を設定

```
gemspec.metadata["msys2_mingw_dependencies"] = "cairo"
```

[MSYS2 library dependencies - For gem developers - onclick/rubyinstaller2 Wiki](#)

# native-package-installer

rcairo case:

```
require "pkg-config"
require "native-package-installer"
unless PKGConfig.have_package("cairo")
  # Install cairo from packaging system automatically
  unless NativePackageInstaller.install(:arch_linux => "cairo",
                                        :debian => "libcairo2-dev",
                                        :homebrew => "cairo",
                                        :macports => "cairo",
                                        :redhat => "cairo-devel")

    exit(false)
  end
  exit(false) unless PKGConfig.have_package("cairo")
end
```

# Stop fat gem problem 3

fat gemをやめたときの問題3

Bundler's  
dependency  
resolution  
may fail

Bundlerの依存関係解決が失敗するかもしれない

# Bundler may use wrong dependency

Bundlerは間違った依存情報を使ってしまふかもしれない

```
git clone https://github.com/rails/rails.git
cd rails
bundle install
bundle update # Error
```

[Bundle Update Fails for Rails 6.1 if Ruby >= 2.6 · Issue #37224 · rails/rails](#)

# Actual

```
Bundler could not find compatible versions for gem "ruby":
```

```
In Gemfile:
```

```
  ruby
```

```
mysql2 (~> 0.5) was resolved to 0.5.2, which depends on  
  ruby (< 2.6) x64-mingw32
```

```
mysql2 (~> 0.5) was resolved to 0.5.2, which depends on  
  ruby (< 2.6) x86-mingw32
```

```
nokogiri (>= 1.8.1) was resolved to 1.10.4, which depends on  
  ruby (>= 2.3) x64-mingw32
```

```
nokogiri (>= 1.8.1) was resolved to 1.10.4, which depends on  
  ruby (>= 2.3) x86-mingw32
```

```
rails was resolved to 6.1.0.alpha, which depends on
```

# Details

詳細

# Omit

省略

# How to resolve this?

## 解決法は？

✓ I've been fixed this  
直しておいた

✓ [rubygems/bundler#7522](https://github.com/rubygems/bundler/issues/7522)

✓ If you're interested in fixing problems in upstream, ClearCode is a good corporation:  
問題をアップストリームで直すことが好きな人はクリアコードといういい会社があるよ！

<https://www.clear-code.com/recruitment/>

✓ Use Bundler 2.2.0 or later (not released yet)  
未リリースだけどBundler 2.2.0以降を使えば解決する

# Wrap up

## まとめ

- ✓ Fat gem was useful until Ruby 2.3  
Ruby 2.3まではfat gemは有用だった
- ✓ We can stop using fat gem now!  
今はfat gemをやめられる!



# Wrap up

まとめ

Thanks fat gem!  
Goodbye fat gem!

ありがとうfat gem!  
さよならfat gem!