GMOペパボ

The Future of library dependency management of Ruby

The integration for package ecosystem

Hiroshi SHIBATA / GMO Pepabo, Inc.

2020.01.16 Ginza Rails #10

self.introduce

GMOペパボ



Hiroshi SHIBATA @hsbt

Executive Officer VP of Engineering
Technical Director
at GMO Pepabo, Inc. @pepabo

https://www.hsbt.org

ruby / ruby

Unwatch 1.2k | Unstar 16.5k | Fork 4.4k

Code | Pull requests 147 | Actions | Security | Insights | Settings

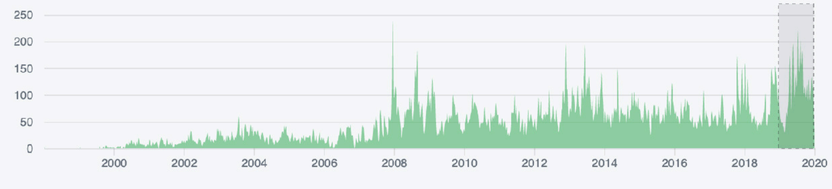
- Pulse
- **Contributors**
- Community
- Traffic
- Commits
- Code frequency
- Dependency graph
- Network
- Forks

- People

Dec 25, 2018 – Dec 25, 2019

Contributions: Commits

Contributions to master, excluding merge commits



#1 nobu
1,436 commits 28,109 ++ 28,019 --



#2 k0kubun
554 commits 6,294 ++ 4,667 --



#3 matzbot
509 commits 3,843 ++ 3,843 --



#4 aycabta
379 commits 37,001 ++ 20,174 --



#5 hsbt
356 commits 114,879 ++ 37,992 --



#6 mame
318 commits 5,781 ++ 3,673 --



#7 jeremyevans
300 commits 16,672 ++ 9,646 --



#8 znz
275 commits 1,744 ++ 3,167 --



No.5

rubygems / rubygems

Used by 190 | Unwatch 144 | Unstar 2.3k | Fork 1.1k

Code | Issues 86 | Pull requests 36 | Actions | Security | Insights | Settings

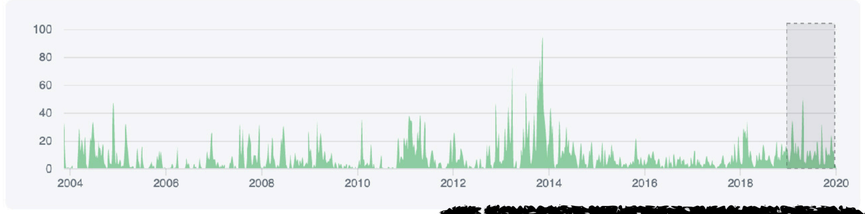
- Pulse
- **Contributors**
- Community
- Traffic
- Commits
- Code frequency
- Dependency graph
- Network
- Forks

- People

Dec 25, 2018 – Dec 25, 2019

Contributions: Commits

Contributions to master, excluding merge commits



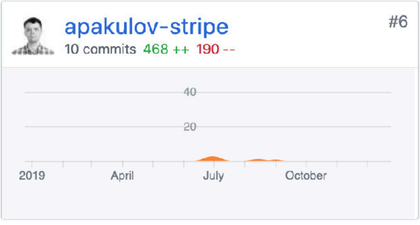
















No.2

bundler / bundler

Sponsor | Used by 149k | Unwatch 172 | Unstar 4.8k | Fork 2.1k

Code | Issues 183 | Pull requests 39 | Actions | Security | Insights | Settings

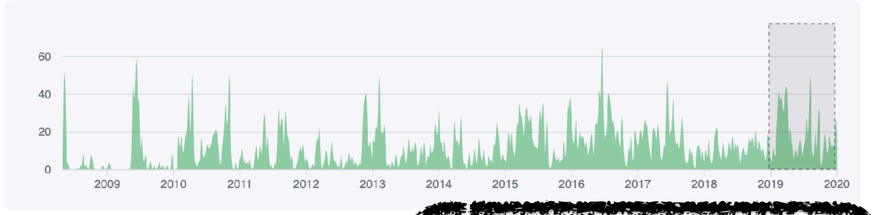
- Pulse
- **Contributors**
- Community
- Traffic
- Commits
- Code frequency
- Dependency graph
- Network
- Forks

- People

Dec 25, 2018 – Dec 25, 2019

Contributions: **Commits**

Contributions to master, excluding merge commits



















No.2

1.

What are RubyGems and Bundler?

What’s rubygems?

RubyGems is a package management framework for Ruby.

- **rubygems/rubygems.org:**
  - **The Ruby community's gem host.**
  - **rubygems.org is maintain by infrastructure team of rubygems. It is different team from rubygems cli team.**

- **rubygems/rubygems:**
  - **Command line tool of rubygems**
  - **Rubygems are created by Seattle.rb**



What's new in RubyGems 3.1

- I released RubyGems 3.1 at 16 Dec 2019
  - https://blog.rubygems.org/2019/12/16/3.1.0-released.html
  - This version dropped the compatibility code of Ruby 1.8 and 1.9
  - Bundle Bundler 2.1
  - Multifactor authentication for yank command



What’s Bundler?

- The vendoring tool of Ruby.
- RubyGems couldn’t care dependency of Ruby libraries and isolate version managing with ruby process.
- Bundler can do them with `Gemfile`

```
# frozen_string_literal: true

source "https://rubygems.org"

git_source(:github) { |repo| "https://github.com/#{repo}.git" }

gemspec

# We need a newish Rake since Active Job sets its test tasks' descriptions.
gem "rake", ">= 11.1"
```



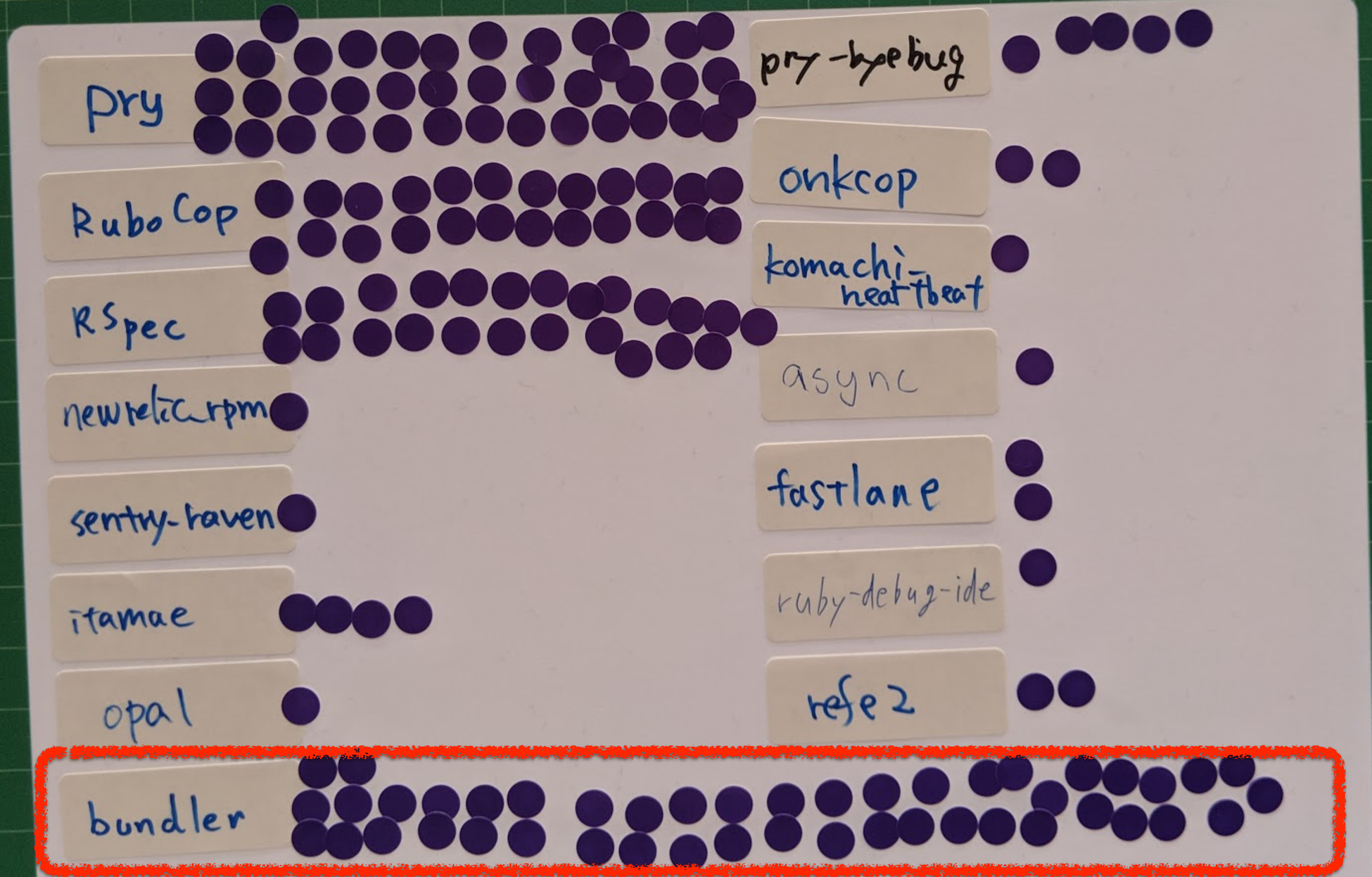
What's new in Bundler 2.1?

- Bundler 2.1 has been released at 15 Dec 2019.
  - https://github.com/bundler/bundler/releases/tag/v2.1.4
  - There is no incompatible feature from Bundler 2.0.x(maybe…)
  - Cleanup the compatibility code for Ruby 1.8 and 1.9
  - Tuned on the deprecation warnings for Bundler 3



最初に入れるgemは？※railsを除く3つまで

What is the gem to install first? ※3 other than rails



Merged Bundler into ruby core 😤

Bundler Integration on RubyGems 3.1

- RubyGems always uses Bundler resolver for gem dependencies
- If you used the Ruby 2.5, You can enabled it with only `gem update —system`

```
ENV["BUNDLE_GEMFILE"] ||= File.expand_path(path)
require 'rubygems/user_interaction'
Gem::DefaultUserInteraction.use_ui(ui) do
  require "bundler"
  @gemdeps = Bundler.setup
  Bundler.ui = nil
  @gemdeps.requested_specs.map(&:to_spec).sort_by(&:name)
end
```



The current behavior of the bundled bundler

- Bundler has been integrated with default gems like json, psych.
- The upstream is https://github.com/bundler/bundler. I backport the released/developed version to ruby repository.

```
benchmark (default: 0.1.0)
bigdecimal (default: 2.0.0)
bundler (default: 2.1.2)
cgi (default: 0.1.0)
csv (default: 3.1.2)
date (default: 3.0.0)
delegate (default: 0.1.0)
did_you_mean (default: 1.4.0)
etc (default: 1.1.0)
fcntl (default: 1.0.0)
fiddle (default: 1.0.0)
fileutils (default: 1.4.1)
…
```



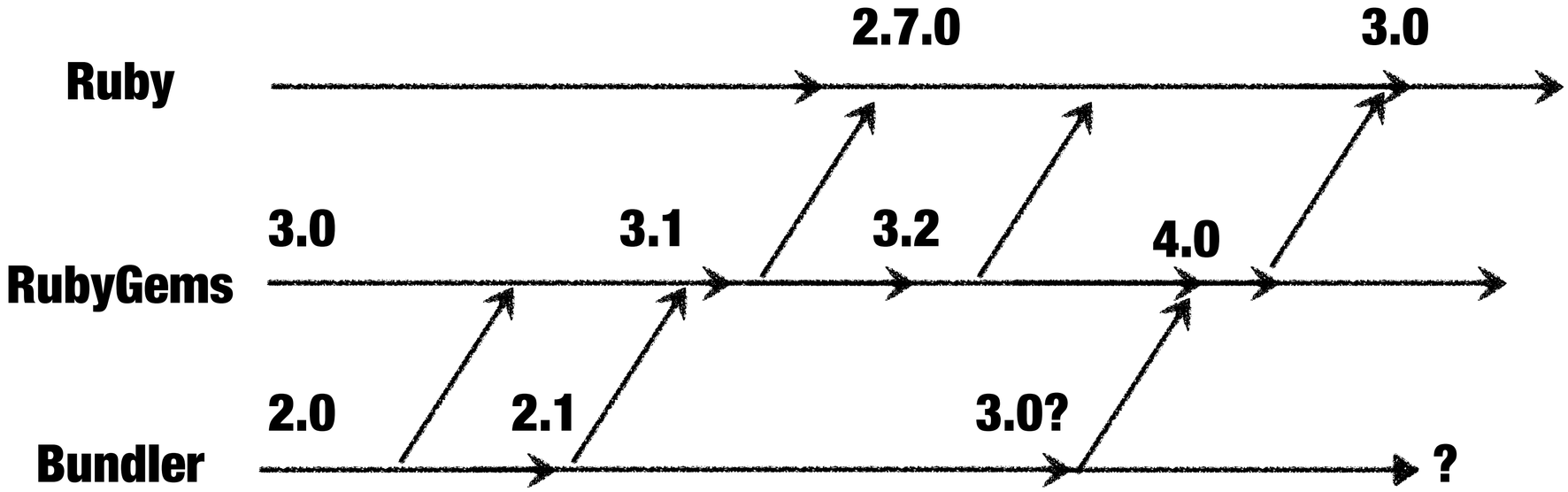
2.

The roadmap for Ruby 3

20/40min

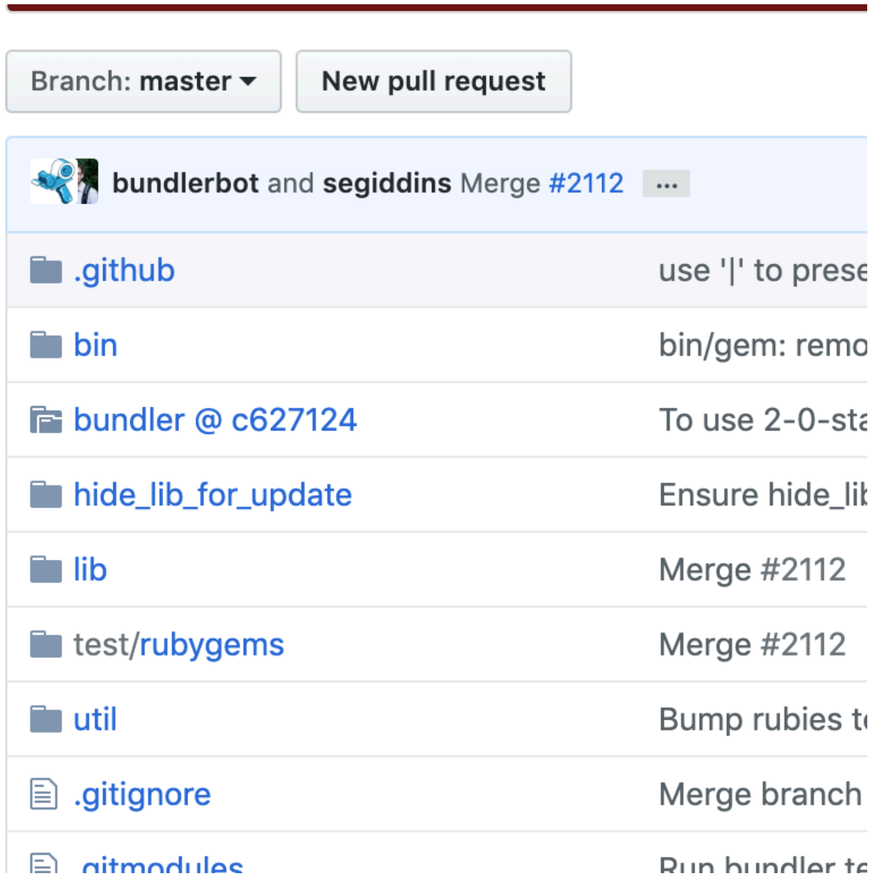
Bump up RubyGems/Bundler

- We will merge into RubyGems 3.2 into Ruby 2.8.0-dev. After that, RubyGems 4.0 will be merge Ruby 3.



RubyGems/Bundler integration

- Now, We put the bundler as submodule in rubygems repository.
- We will move the canonical repository of bundler to rubygems org or rubygems/rubygems.



Dependency Resolver incompatible

- RubyGems 2.x and 3.x uses Molinillo-0.5.7

```
~/D/g/r/rubygems (master) > ls lib/rubygems/resolver/molinillo/lib/molinillo
delegates           dependency_graph.rb gem_metadata.rb     resolution.rb       state.rb
dependency_graph    errors.rb           modules             resolver.rb
```

- Bundler 1.x and 2.x also uses Molinillo-0.6.4

```
~/D/g/b/bundler (master) > ls lib/bundler/vendor/molinillo/lib/molinillo
compatibility.rb    dependency_graph    errors.rb           modules             resolver.rb
delegates           dependency_graph.rb gem_metadata.rb     resolution.rb       state.rb
```

- These are different versions and behavior of dependency resolver.



Make conservative option as default

- We got the installation time when already installed gems.

```
~ > gem i rails
Successfully installed rails-5.2.0
1 gem installed
```

- To use conservative is ignore re-install action.

```
~ > gem i rails ——conservative
~ >
```



Make `--user-install` as default

- RubyGems 4 will install the all gems to `~/.gem` maybe.
- Pros: Ruby in linux distribution has many of FAQ for gem installation for using `sudo`. This change resolve this issues.
- Cons: Ruby version manager like rbenv is not support it. And RubyGems have a lot of issues related this.



Activation issues about default gems

- You couldn’t use the specified version of default gems like json when RubyGems/Bundler activated them.
- When rubygems uses json-2.1.0, You couldn’t use json 1.8.x. Because ruby gems and rubygems.org never uses JSON format.
- We can resolve it with `vendoring` approach. But json, psych, and openssl is C extension library.



The feature flag for Bundler 3

```
    settings_flag(:allow_bundler_dependency_conflicts) { bundler_3_mode? }
    settings_flag(:allow_offline_install) { bundler_3_mode? }
    settings_flag(:auto_clean_without_path) { bundler_3_mode? }
    settings_flag(:auto_config_jobs) { bundler_3_mode? }
    settings_flag(:cache_all) { bundler_3_mode? }
    settings_flag(:default_install_uses_path) { bundler_3_mode? }
    settings_flag(:deployment_means_frozen) { bundler_3_mode? }
    settings_flag(:disable_multisource) { bundler_3_mode? }
    settings_flag(:forget_cli_options) { bundler_3_mode? }
    settings_flag(:global_gem_cache) { bundler_3_mode? }
    settings_flag(:only_update_to_newer_versions) { bundler_3_mode? }
    settings_flag(:path_relative_to_cwd) { bundler_3_mode? }
    settings_flag(:plugins) { @bundler_version >= Gem::Version.new("1.14") }
    settings_flag(:print_only_version_number) { bundler_3_mode? }
    settings_flag(:setup_makes_kernel_gem_public) { !bundler_3_mode? }
    settings_flag(:skip_default_git_sources) { bundler_3_mode? }
    settings_flag(:specific_platform) { bundler_3_mode? }
    settings_flag(:suppress_install_using_messages) { bundler_3_mode? }
    settings_flag(:unlock_source_unlocks_spec) { !bundler_3_mode? }
    settings_flag(:update_requires_all_flag) { bundler_4_mode? }
    settings_flag(:use_gem_version_promoter_for_major_updates) { bundler_3_mode? }

    settings_option(:default_cli_command) { bundler_3_mode? ? :cli_help : :install }
```



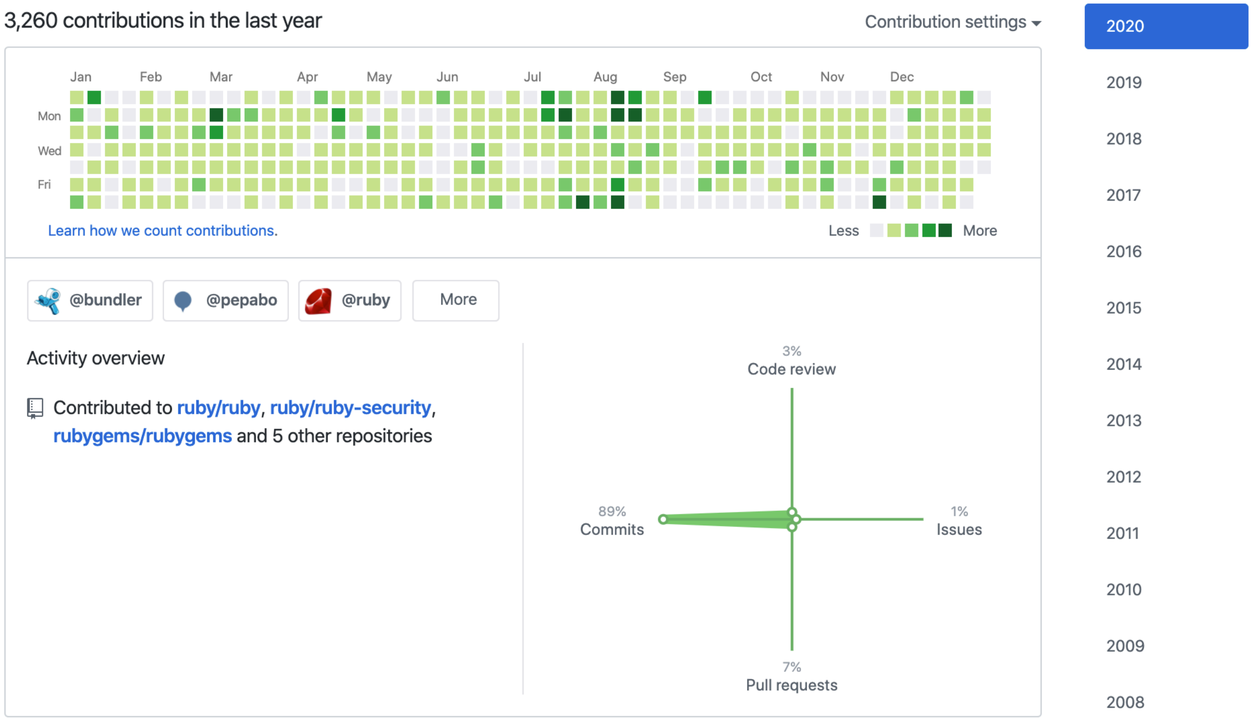
3.

How maintain the OSS with Daily work

I'm NOT
the full-time OSS developer

The detail of hsbt's OSS work

- My mainly "salary work" is only top-level management for the engineering, product and people.
- The management is async work. So I write the code in waiting to the result of my management.
- I did contribute 10/day with code, issue triage and others.



Why do I work with OSS?

- The valuable OUTPUT have a big influence. So, OSS has a power of influence rather than closed-software.
- OSS teach to me the collaborative mindset. So, OSS is NOT code falling on the road. You can see the people throw the OSS.
- OSS is Fun.



How it works with everyday?

Ignore Burnout

- OSS is hard
  - “Help! my code is broken after version-up.”, “Error”, “Your code is slow”…🔥🔥🔥
- But, OSS is easy
  - You can ignore the issue yourself and go to the bedroom 🛏️
  - You don't have a responsibility of resolve it for them. Say “Thank you! I will work it in the future.” Or “We always welcome your patch!”



Courage for the OSS maintenance

- Close issue or pull-request with not-want feature at first.
  - Pull-request says “You should maintain my dirty code in the feature”

- Ignore the suggestion of code-style and trivial changes
  - Ex1. Added rubocop rule, Fixed indentation, etc.
  - Ex2. Refactor



How do contribute sustainable

- Do not care your pull-request regularly
  - When You need to apply your patch with upstream, You can fork it 😁
  - Remember the previous slide that is “Courage for the OSS maintenance”

- If you want to merge your pull-request definitely.
  - You should write the use-case, expected/actual behavior, coding-style, benchmark result, commit message every pull-request and everyday.



Survive with the Social Coding

- OSS is not only CODE. It mainly focus the human being after GitHub.
- OSS maintainers especially me consider your being more than your code.



Ruby is designed to make programmers happy.

Yukihiro Matz Matsumoto