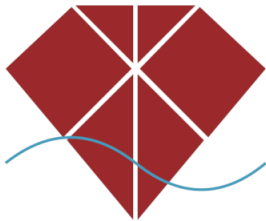


dRuby 入門者による あなたの身近にある dRuby 入門

@makicamel
大阪 Ruby 会議 04
2024.08.24



Osaka RubyKaigi 04

自己紹介

- @makicamel / 川原万季
- Ruby 💎 とビール 🍺 とお酒が好き
- 好きな VR ゲームは BeatSaber 🗡️
- (株)アンドパッド 🧑‍🔧





dRuby

- distributed ruby
 - 分散 Ruby
- 分散オブジェクトシステムを実現するライブラリ
- 1999 年初出
- 関将俊さん作



dRuby

- 1999 年初出
- <https://gist.github.com/seki/5713863>

1st druby. [ruby-list:15406]

drb.rb


Raw

```
1 drb.rb
2 #!/usr/local/bin/ruby
3 =begin
4   Tiny distributed Ruby --- dRuby
5   DRb --- dRuby module.
6   DRbProtocol --- Mixin class.
7   DRbObject --- dRuby remote object.
8   DRbConn ---
9   DRbServer --- dRuby message handler.
10 =end
11
12 require 'socket'
13 require 'marshal'
14
15 module DRb
16   def start_service(uri, front=nil)
17     @uri = uri.to_s
18     @front = front
19     @server = DRbServer.new(@uri)
20     @thread = @server.run
21   end
22   module function :start_service
```



dRubyによる分散・Webプログラミング / 関将俊
<https://www.ohmsha.co.jp/book/9784274066092/>

dRuby

- distributed ruby
 - 分散 Ruby
 - 分散オブジェクトシステムを実現するライブラリ
 - 1999 年初出
 - 関将俊さん作
- 

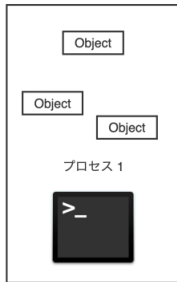
分散オブジェクトシステム

- とは？

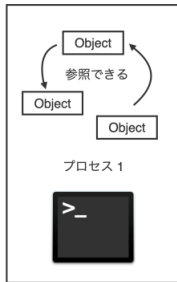
分散オブジェクトシステム

- プロセスやネットワークを超えてメッセージを送る
 - 他のプロセスのオブジェクトのメソッドを呼び出せる

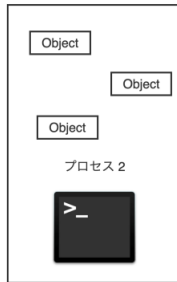
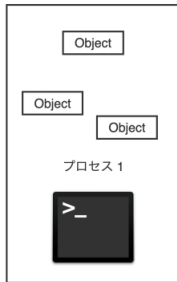
分散オブジェクトシステム



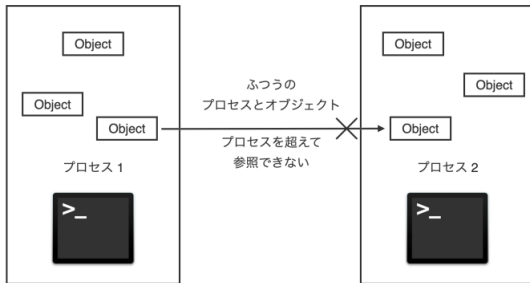
分散オブジェクトシステム



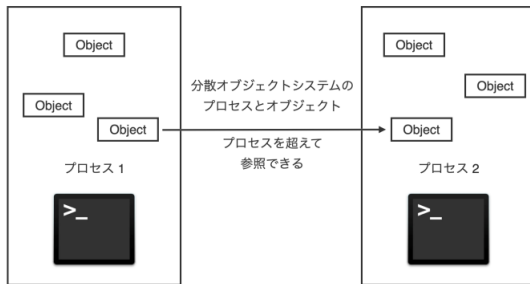
分散オブジェクトシステム



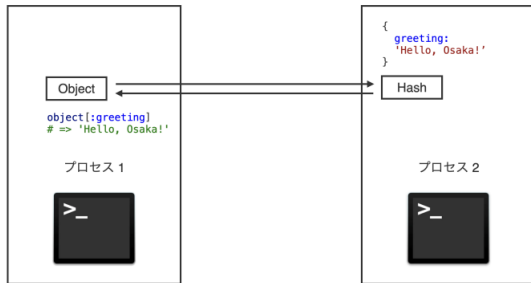
分散オブジェクトシステム



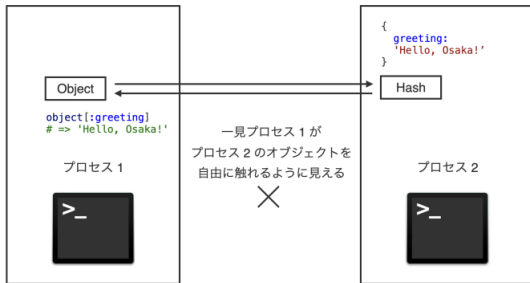
分散オブジェクトシステム



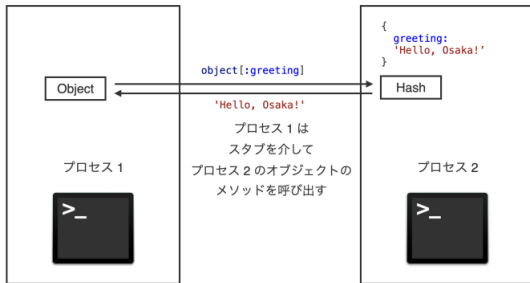
分散オブジェクトシステム



分散オブジェクトシステム



分散オブジェクトシステム



分散オブジェクトシステム

- プロセスやネットワークを超えてメッセージを送る
 - 他のプロセスのオブジェクトのメソッドを呼び出せる
- かんたんにサーバが作れる

夢が広がる！！



分散オブジェクトシステム

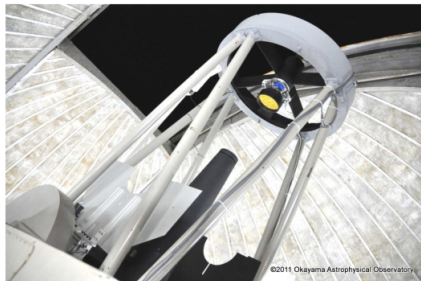
```
# terminal
$ irb -r drb
rabbit = DRbObject.new_with_uri 'druby://127.0.0.1:10101' # rabbitのdRubyサーバのuri
rabbit.title
# => "dRuby 入門者による\nあなたの身近にある\ndRuby 入門"
%w[Rabbit最高 🐇🐇🐇🐇🐇🐇🐇🐇 dRubyすごい 大阪たのしい🍷].each { rabbit.append_comment _1 }
z
# => [...]
rabbit.toggle_fullscreen
```

スタブを介してメッセージを送ることで Rabbit を操作できる

分散オブジェクトシステム

- リアルワールド dRuby 活用例

分散オブジェクトシステム



岡山天体物理観測所広視野赤外カメラの製作 / 柳澤 顕史

https://www.astr.tohoku.ac.jp/~akiyama/astroinst2015/20151207_yanagisawa.pdf

Rinda in the real-world embedded systems. / 関将俊

https://rubykaigi.org/2020-takeout/presentations/m_seki.html

分散オブジェクトシステム

● ● ● Point of Care Testing

- Rapid tests for the Respiratory infection diseases such as SARS-CoV-2, Influenza, RS virus
- Next Generation
- +UI, Parallel processing



dRuby in the real-world embedded systems. / 関将俊 園川龍也

<https://www.druby.org/seki-RK2021.pdf>

https://rubykaigi.org/2021-takeout/presentations/m_seki.html

分散オブジェクトシステム



Scaling Twitter / Blaine Cook

<https://www.slideshare.net/slideshow/scaling-twitter/41197>

分散オブジェクトシステム

■ すごすぎる

もうちょっと
身近な例ありませんか



dRuby 入門者によるあなたの身近にある dRuby 入門



わたしたちの身近にある **dRuby**

- RSpec
- ActiveSupport
- Rabbit
- るりま

わたしたちの身近にある **dRuby**

- **RSpec** 
- ActiveSupport
- Rabbit
- るりま

RSpec

■ drb オプション

```
$ rspec --help  
-X, --[no-]drb  
    --drb-port PORT
```

Run examples via DRb.
Port to connect to the DRb server.

RSpec

- アプリケーションを dRuby プロセスに読み込んでテストサーバにする
- rspec 実行時にテストサーバでテスト実行する



RSpec



① dRuby サーバ起動
→ テストサーバ



RSpec



② `rspec -drb`



RSpec



③ テストサーバにテスト依頼



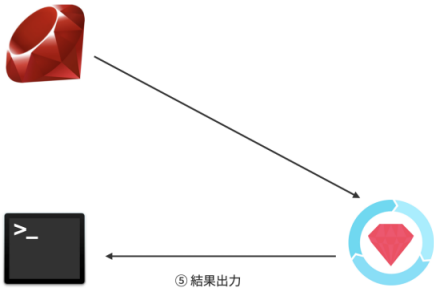
RSpec



④ テスト実行



RSpec



RSpec

- テストサーバ
- プリローダ
 - アプリケーション起動時間を省略
- e.g. spork
 - A DRb server for testing frameworks
 - ※ 最終コミットは 2014 年
 - 2024 年現在はまだあまり使われていないかも

spork
<https://github.com/sporkrb/spork>

わたしたちの身近にある **dRuby**

- RSpec
- **ActiveSupport** 
- Rabbit
- るりま

ActiveSupport

- ActiveSupport::TestCase#parallelize
- テストの並列化
- 使用マシンのコア数分プロセスをフォーク

ActiveSupport::TestCase

<https://api.rubyonrails.org/classes/ActiveSupport/TestCase.html#method-c-parallelize>



ActiveSupport

- rails new で test_helper.rb を生成
 - デフォルトで parallelize が記述されている
- 何も意識しなくてもテストが並列実行される

```
# test/test_helper.rb
module ActiveSupport
  class TestCase
    parallelize(workers: :number_of_processors)

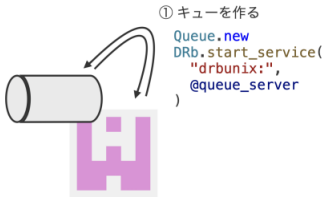
    fixtures :all
  end
end
```

ActiveSupport

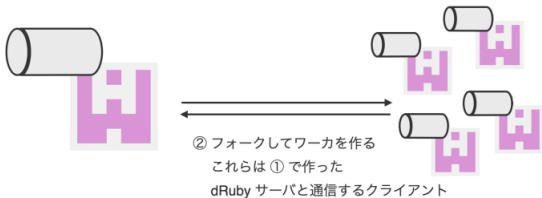
- ActiveSupport::TestCase#parallelize の並列実行のしくみ



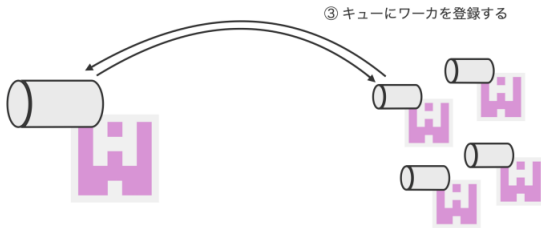
ActiveSupport



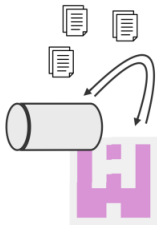
ActiveSupport



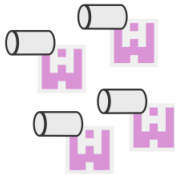
ActiveSupport



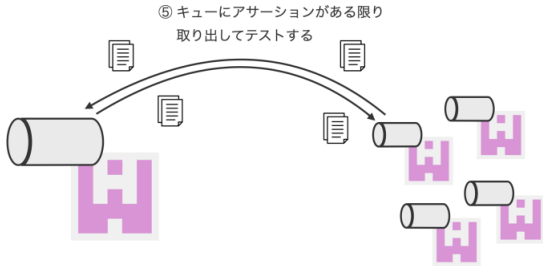
ActiveSupport



④ キューにアサーションを詰める



ActiveSupport

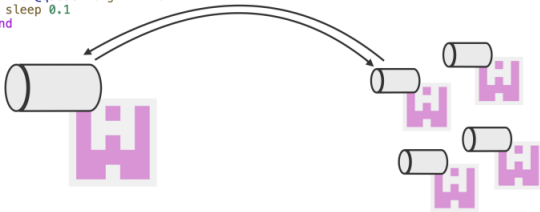


ActiveSupport

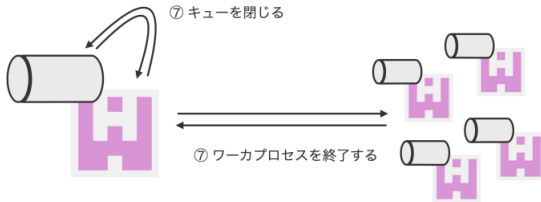
キューが空になるまで
待つ

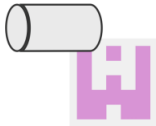
```
while @queue.length != 0  
  sleep 0.1  
end
```

⑥ キューが空になったら
キューからワーカを削除する



ActiveSupport






⑧ テスト結果を返す

ActiveSupport

- 並列実行のしくみ
 - アサーションの管理を dRuby に任せる
 - フォークしたプロセスが各個アサーションを引き出しテスト実行

テスト結果の管理は親プロセスの仕事



わたしたちの身近にある **dRuby**

- RSpec
- ActiveSupport
- **Rabbit** 
- るりま

Joy

24



89

Powered by Rabbit 3.0.0

Contribute to Ruby - Yukihiro "Matz" Matsumoto

<https://speakerdeck.com/matz/contribute-to-ruby-rubykaigi-2022?slide=25>

https://rubykaigi.org/2022/presentations/yukihiro_matz.html

Rabbit

- Rubyist 御用達プレゼンテーションツール
 - うさぎとかめでお馴染み
- 須藤功平さん作

Rabbit

<https://rabbit-shocker.org/ja/>

<https://github.com/rabbit-shocker/rabbit>



Rabbit

■ デフォルトで dRuby サーバを立ち上げる

```
class Rabbit::Command::Rabbit
  def do_display
    # ...
    front = make_front(canvas)
    setup_druby(front) if @options.use_druby
    # ...
  end

  def setup_druby(front)
    require "drb/drb"
    begin
      DRb.start_service(@options.druby_uri, front)
    # ...
    end
  end
end
```

Rabbit

```
# terminal
$ irb -r drb
rabbit = DRbObject.new_with_uri 'druby://127.0.0.1:10101' # rabbitのdRubyサーバのuri
rabbit.title
# => "dRuby 入門者による\nあなたの身近にある\nndRuby 入門"
%w[Rabbit最高 🐇🐇🐇🐇🐇🐇🐇🐇 dRubyすごい 大阪たのしい🍷].each { rabbit.append_comment _1 }
rabbit.available_interfaces
# => [...]
rabbit.toggle_fullscreen
```

スタブを介してメッセージを送ることで Rabbit を操作できる

Rabbit

- API 提供

Rabbit API

■ RabbiRack

- Web ブラウザから Rabbit を操作するツール

```
get "/pages/next" do
  @rabbit.move_to_next_if_can
  haml :index
end
```

RabbiRack

<https://rabbit-shocker.org/ja/rabbirack>

<https://github.com/rabbit-shocker/rabbirack>

※ macOS では <https://github.com/rabbit-shocker/rabbit/commit/2b2ce805d98fd972d0a288ac7df944e53d48bbbf> の適用が必要

Rabbit API

■ Rabbiter

- Twitter からツイートを収集し、コメントとして Rabbit に流しこむツール

```
rabbit = DRbObject.new_with_uri(options.rabbit_uri)
client = Rabbiter::Client.new(logger)
client.start(*options.filters) do |status|
  # ...
  comment = "@#{status.user.screen_name}: #{clean_text(status, options)}"
  rabbit.append_comment(comment)
end
```

Rabbiter

<https://rabbit-shocker.org/ja/rabbiter>

<https://github.com/rabbit-shocker/rabbiter>

※ Twitter (現 X) が壊れているので 2024 年 8 月現在は動かさない

わたしたちの身近にある **dRuby**

- RSpec
- ActiveSupport
- Rabbit
- るりま 

オブジェクト指向スクリプト言語 Ruby リファレンスマニュアル

- Ruby オフィシャルサイト <https://www.ruby-lang.org/ja/> 
- version 3.3 対応リファレンス
- 原著: まつもとゆきひろ
- 最新版URL: <https://www.ruby-lang.org/ja/documentation/> 

[\[edit\]](#)

使用上の注意

組み込みクラスのリファレンスはほぼ揃っています。標準添付ライブラリのリファレンスは一部未完成です。それ以外のドキュメントについては、まだまだ書き直しが必要です。

目次

- はじめに
- コマンド
- Rubyの起動
- 環境変数

Ruby 言語仕様

Ruby でのオブジェクト:

- オブジェクト
- クラス

プロセスの実行:

- Ruby プログラムの実行
- 終了処理
- スレッド

るりま

- doctree

- ドキュメント
- <https://github.com/rurema/doctree>

- BitClust

- ドキュメントシステム
- リファレンスデータベースの更新、表示、検索、html 生成など
- <https://github.com/rurema/bitclust>

BitClust

- BitClust データベース
 - Ruby のクラスをふつうに使う
 - rd をコンパイルしたデータを持つ
 - テキストファイルで永続化
 - 人間可読のテキスト
- MySQL でも PostgreSQL でも Redis でもないデータベース
- SQL も O/R マッパーも要らない

BitClust

■ refe

- るりまのエントリを引くためのコマンドライン用ツール
- デーモン化
 - BitClust データベースをフロントオブジェクトとして dRuby サーバを起動

```
class BitClust::Server
  def listen(url, foreground = false)
    # ...
    WEBrick::Daemon.start unless foreground
    DRb.start_service url, @db
    DRb.thread.join
  end
end
```

BitClust

- データベースとしての Ruby プロセス
 - PORO でデータ操作
 - ファイルに永続化
- ふつうの Ruby プログラミングの延長線上でデータベースサーバを作れる

わたしたちの身近にある **dRuby**

- プリローダ
- 並列
- API
- データベース

夢が広がる！！



たとえば明日から **dRuby** を使うとしたら



たとえば明日から **dRuby** を使うとしたら

- PoC

- Web サーバを dRuby で立ててみる
- DB サーバを dRuby で立ててみる

- 手に馴染む道具で検討して固まったら本実装に入るのも手かも

たとえば明日から **dRuby** を使うとしたら

■ PoC

- Web サーバを dRuby で立ててみる
- DB サーバを dRuby で立ててみる
- 手に馴染む道具で検討して固まったら本実装に入るのも手かも

分散オブジェクトシステムに
入門したくなったら





dRubyによる分散・Webプログラミング / 関将俊
<https://www.ohmsha.co.jp/book/9784274066092/>

ご清聴

ありがとう

ございました