

# 全文検索エンジン Groongaへの mrubyの組み込み

須藤功平

Groongaプロジェクト

第7回フクオカRuby大賞  
2015-01-23





# 説明すること

1. 特徴
2. Rubyを適用した箇所
3. Rubyを活用したことによる  
効果、社会に及ぼす影響



# (1) 特徴

mrubyの使い方が  
少し違う



# 使い方の分類

主体 対象者	mruby	C
ユーザー 向け	1	2
開発者 向け	3	4



# 分類の補足

- 主体
  - メインの処理を実装する言語
  - あるいは
  - プログラムのエントリーポイントを実装する言語
- 対象者
  - Rubyを書く人はだれか



# よくある使い方

主体 対象者	mruby	C
ユーザー 向け	1	2
開発者 向け	3	4



# Groongaの使い方

主体 対象者	mruby	C
ユーザー 向け	1	2
開発者 向け	3	4



# よくある使い方 (1)

- ユーザー向け・mruby主体  
ユーザーがRubyを書く・mrubyでメインの処理を実装
  - 拡張性が高いアプリ
  - PCで動かすならCRubyでもよい  
CRuby + バインディング
  - 単体で配布可能なバイナリーを作れることが便利





# よくある使い方 (2)

## ■ ユーザー向け・C主体

ユーザーがRubyを書く・Cでメインの処理を実装

### ■ プラグイン・設定

■ C側がマルチスレッドや  
マルチプロセスでも組み込みやすい

### ■ CRubyだと難しい・面倒

シグナル・RUBY\_INIT\_STACK...



# よくある使い方 (3)

## ■ 開発者向け・mruby主体

開発者がRubyを書く・mrubyでメインの処理を実装

- 組み込み環境向けアプリケーション
- 消費リソースが少なくてうれしい
- 開発速度があがる
- CRuby感覚で使うと機能が少なくてツライ



# Groongaの使い方

主体 対象者	mruby	C
ユーザー 向け	1	2
開発者 向け	3	4



# Groongaの使い方

## ■ 開発者向け・C主体

開発者がRubyを書く・Cでメインの処理を実装

- アプリ内のコア機能の一部を実装
- 開発速度があがる
- mruby対応がそこそこ大変  
バインディングの開発・ビルドシステムへの統合・  
テストの仕組み…

# 開発者向け・C主体の使い方

- 使い所は難しい
  - 仕組み作りを乗り越えられれば有用
  - 仕組みできる→**開発速度はあがる**
- 難易度も高い
  - プロジェクト依存部分大きい
  - Groongaの事例から一般化できるといいなあ

# (2) Rubyを適用した箇所

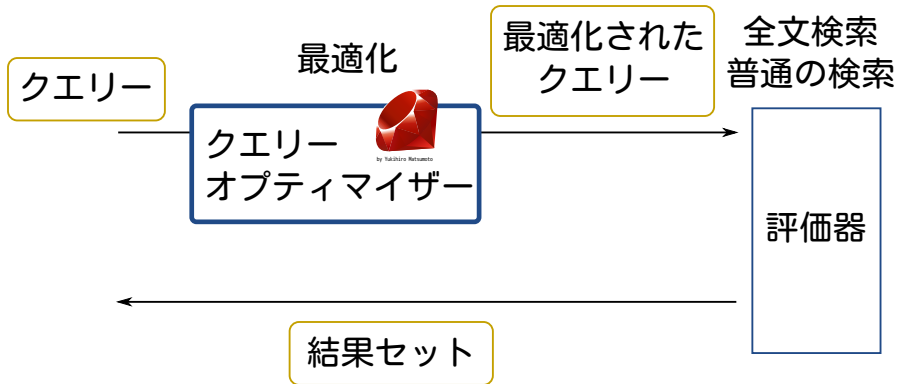
1. 特徴
2. **Rubyを適用した箇所**
3. Rubyを活用したことによる  
効果、社会に及ぼす影響



# コア機能の実装

- クエリーオプティマイザー
- 予定：式のJIT化

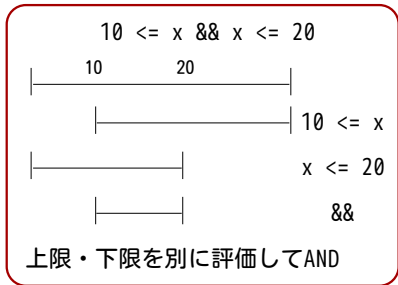
# クエリー最適化



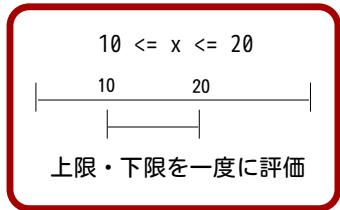




# 例



最適化



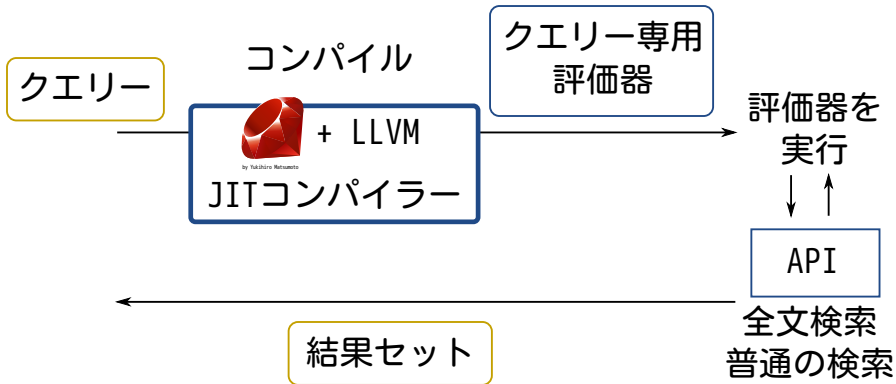


# mrubyは割にあうか

- 最適化→10倍以上高速化も可
  - C→mrubyの速度低下は影響小
- C→mrubyで開発速度向上
  - 課題：GDBが使えない
  - →ppをmrubyに移植してなんとか…



# 予定：式のJIT化





## (3) 影響

1. 特徴
2. Rubyを適用した箇所
3. Rubyを活用したことによる  
効果、社会に及ぼす影響



# 影響 (1)

- 「コア機能をRubyで実装」という使い方の実例を示した
  - フリーソフトウェアなので他のプロダクトは参考にできる
  - 仕組み作りをできれば割にあう
  - → 「技術支援」という仕事をしたい



# 影響 (2)

- mrubyがよくなる
  - 開発中に見つけた問題は直してパッチを送っている



# まとめ

1. 特徴
  - 類をみないmrubyの使い方
2. Rubyを適用した箇所
  - クエリーオプティマイザー
3. Rubyを活用したことによる効果、社会に及ぼす影響
  - 参考にできる・mrubyがよくなる