

Embed Ruby

アプリケーションへの
Rubyインタープリターの組み込み

須藤功平

株式会社クリアコード

東京Ruby会議11
2016-05-28

Speaker's award

Continuous development award

受賞者

cedLemo

Continuous development award

受賞理由

2015年1月から継続的に
Ruby-GNOME2の開発に
参加しているから

一発すごい改善をした人よりも
地味でも継続的に改善している人を評価したい

Continuous development award
cedlemo

宣伝

OSS Gate

OSS Gate

OSS開発に
参加する人を
増やす取り組み

背景

- ✓ OSS利用は当たり前になった
 - ✓ →OSS利用者増加
- ✓ 開発参加者も増えるといいな
 - ✓ →OSS増加

OSS Gate

OSS開発参加

- ✓ すごい改善じゃなくていい
- ✓ バグレポートとかでいい
 - ✓ typoを見つけました！とかでもいい
 - ✓ サンプルを更新とかでもいい

OSS Gate

OSS Gate参加の動機

人それぞれでいい

OSS Gate

私の動機

- ✓ ユーザーが自由に使える
ソフトウェアが増えるといいな
- ✓ 自由に使える例：
 - ✓ コードを読んで学習できる
 - ✓ 今日聞いた話の実装を確認できる！

OSS Gate

興味ある？

- ✓ 興味？（重要：動機不問）
 - ✓ OSS開発に参加したい！
 - ✓ OSS開発参加者を増やしたい！
- ✓ 5階でワークショップ開催中
 - ✓ 説明や見学は私に一声かけて

OSS Gate

本題

Rubyの組み込み

(CアプリケーションへのRubyインタープリターの組み込み)

動機

- ✓ 柔軟な記述力が欲しい
- ✓ Cの速さが欲しい
- ✓ Ruby以外の言語とも連携したい
 - ✓ Pythonも組み込む
- ✓ なんかカッコいい

Rubyの組み込み

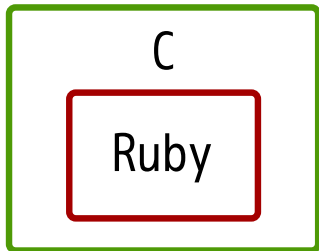
別の実現方法

拡張ライブラリー

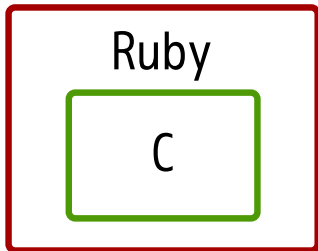
Rubyの組み込み

組み込みと拡張ライブラリー

Ruby組み込み



拡張
ライブラリー



Rubyの組み込み

拡張ライブラリー

- ✓ 実現可能：
 - ✓ 柔軟な記述力が欲しい
 - ✓ Cの速さが欲しい
- ✓ 実現不可能：
 - ✓ Ruby以外の言語とも連携したい
 - ✓ なんかカッコいい感

Rubyの組み込み

実現方法の選び方

- ✓ 基本は拡張ライブラリー
- ✓ すでにあるアプリなら組み込み
- ✓ 選びたい方があるならそっち

Rubyの組み込み

組み込みを選ぶ時の注意

それなりの覚悟が必要

- ✓ 利用例があまりない
 - ✓ 問題遭遇確率が高い
- ✓ 問題遭遇時：
 - ✓ 自分でソースを読んで調べる
 - ✓ 詳しい人に相談
ささださんくできればサポートを強化したい

Rubyの組み込み

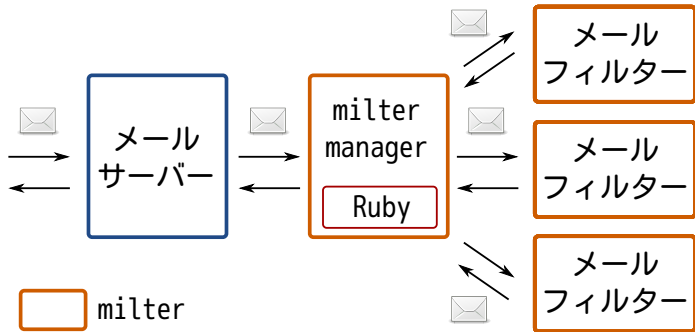
組み込み方を紹介

実例

milter manager

Since 2008

milter manager



milterを管理するmilter
サーバープロセス

Ruby組み込みの実装

- ✓ 初期化
- ✓ fork対応
- ✓ イベントループとシグナル

milter managerへのRubyの組み込み

初期化：GC関連

```
{  
  /* スタックの底を設定 */  
  /* GC時にCのローカル変数に代入されている  
     Rubyのオブジェクトをマークするため */  
  RUBY_INIT_STACK;  
  /* ... */  
}
```

milter managerへのRubyの組み込み

スタックとマーク対象

```
{
  RUBY_INIT_STACK;
  /* ... */
  {
    VALUE object = rb_ary_new(); /* マーク対象 */
  }
}
{
  VALUE object = rb_ary_new(); /* マーク対象外 */
}
```

milter managerへのRubyの組み込み

確認例

```
#define MARKED_P(object) rb_objspace_marked_object_p(object)
{
  RUBY_INIT_STACK;
  {
    VALUE object = rb_ary_new();
    /* GC.start(immediate_sweep: false) */
    printf("%d\n", MARKED_P(object)); /* => 1 */
  }
}
{
  VALUE object = rb_ary_new();
  /* GC.start(immediate_sweep: false) */
  printf("%d\n", MARKED_P(object)); /* => 0 */
}
```

スライドのリポジトリ: [examples/gc.c](#)

GC関連の注意

```
{  
  RUBY_INIT_STACK;  
  /* Rubyのオブジェクトを触る  
     Cのコードはこのブロック内でだけ使うこと */  
  /* Cからのコールバックで  
     Rubyのコードを呼び出すときは注意 */  
}
```

milter managerへのRubyの組み込み

初期化：シグナル関連

```
{  
  RUBY_INIT_STACK;  
  /* シグナルハンドラーを保存 */  
  ruby_init(); /* Rubyがシグナルハンドラーを登録 */  
  /* シグナルハンドラーを復帰 */  
  /* シグナルはアプリで処理したいから */  
}
```

milter managerへのRubyの組み込み

シグナル復帰例

```
{  
  /* 他のシグナルも同様に復帰 */  
  void (*sigint_handler)(int);  
  sigint_handler = signal(SIGINT, SIG_DFL);  
  ruby_init();  
  signal(SIGINT, sigint_handler);  
}
```

milter managerへのRubyの組み込み

初期化：引数の処理

```
{  
  /* ...ruby_init()... */  
  static char *argv_raw[] = {"milter-manager", "-e;"};  
  int argc;  
  char **argv;  
  argc = sizeof(argv_raw) / sizeof(char *);  
  argv = argv_raw;  
  ruby_incpush(/* ... */); /* $LOAD_PATHの設定 */  
  /* 中でいろいろ初期化するのでダミーの引数で呼ぶ */  
  ruby_process_options(argc, argv);  
}
```

milter managerへのRubyの組み込み

初期化：アプリの初期化

```
{  
  /* ...ruby_process_options()... */  
  /* require中に例外が発生してもここで止める */  
  /* ここで止めないと例外を受け取る人がいなくて  
     クラッシュ */  
  rb_protect(/* rb_require("milter/manager") */);  
}
```

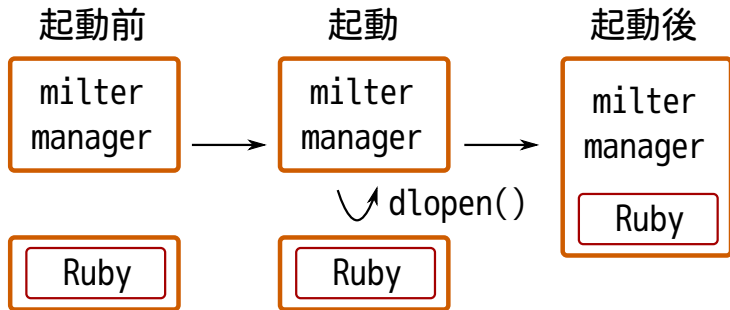
milter managerへのRubyの組み込み

milter manager と Ruby

- ✓ 組み込み処理系の1つ
 - ✓ Pythonも使えるようにしたかった
結局Ruby必須でPython対応はしなかった
- ✓ 起動後にdlopen()で動的にsoを読み込んで組み込み

milter managerへのRubyの組み込み

起動時に動的に組み込み



milter managerへのRubyの組み込み

起動時にso読んで組み込み

```
{
  /* ↓GC用 */
  RUBY_INIT_STACK;
  /* 動的にsoを読んで初期化関数を呼ぶ */
  /* dlopen();
     init = dlsym();
     init(); ←の中でruby_init();とか */
  /* アプリの処理 */
}
```

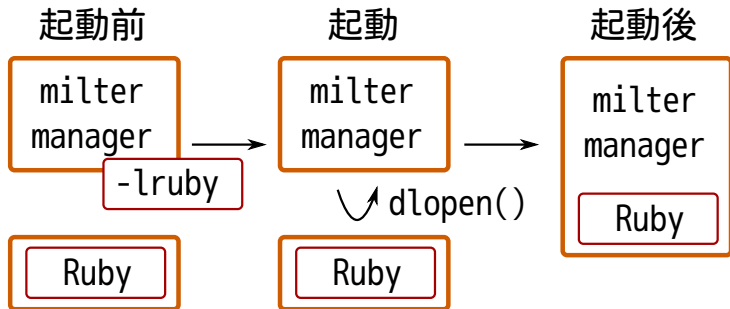
milter managerへのRubyの組み込み

RUBY_INIT_STACK ! ?

```
{
  /* ↓アプリ側で呼ぶの! ? */
  RUBY_INIT_STACK;
  /* 動的にsoを読んで初期化関数を呼ぶ */
  /* dlopen();
     init = dlsym();
     init(); ←の中でruby_init();とか */
  /* アプリの処理 */
}
```

milter managerへのRubyの組み込み

アプリに-lruby...



カッコわるい。。。
milter managerへのRubyの組み込み

Ruby組み込み時の意気込み

- ✓ 本体に組み込む
- ✓ 動的に組み込もうとしない

Ruby組み込みの実装

- ✓ 初期化
- ✓ fork対応
- ✓ イベントループとシグナル

milter managerへのRubyの組み込み

milter manager利用例

- ✓ 大学・企業
 - ✓ ユーザー数：数百～数万人
- ✓ プロバイダー
 - ✓ ユーザー数：数千～数十万人

それなりの性能が必要

milter managerへのRubyの組み込み

性能向上方法

- ✓ CPU
 - ✓ マルチプロセス1択
- ✓ 通信・多同時接続
 - ✓ いろいろ

milter managerへのRubyの組み込み

マルチプロセス

✓ マスタープロセス

a. listen()

b. fork()

✓ ワーカープロセス

a. accept()

b. ↑したクライアントの処理

milter managerへのRubyの組み込み

Ruby組み込みとfork()

- ✓ fork()すると
ワーカープロセスがクラッシュ
- ✓ プロセス終了時とか
- ✓ ヒント：fork()とスレッド

milter managerへのRubyの組み込み

fork()とスレッド

- ✓ 混ぜるな危険
- ✓ Rubyはスレッドを動かしている
 - ✓ 例：タイマースレッド
- ✓ fork時にスレッドのケアが必要

milter managerへのRubyの組み込み

スレッドのケア

```
VALUE rb_pid;  
/* タイマースレッドの後始末とか  
   した上でfork */  
rb_pid = rb_funcall(rb_mKernel,  
                    rb_intern("fork"),  
                    0);  
return NUM2INT(rb_pid);
```

milter managerへのRubyの組み込み

Ruby組み込みの実装

- ✓ 初期化
- ✓ ~~fork~~対応
- ✓ イベントループとシグナル

milter managerへのRubyの組み込み

イベントループとシグナル

- ✓ 気にしなくてよい
 - ✓ アプリがシグナルを処理するから
- ✓ 拡張ライブラリーなら対応必要
 - ✓ イベントループ中にシグナル発生
 - ✓ →すぐにイベントループを抜ける

milter managerへのRubyの組み込み

Rubyの組み込みのまとめ

- ✓ Rubyを組み込む実装方法を紹介
- ✓ 動的組み込みは諦めろ
- ✓ fork時はRubyのforkを使う

Rubyの組み込み

mrubyの組み込み

実例

Groonga

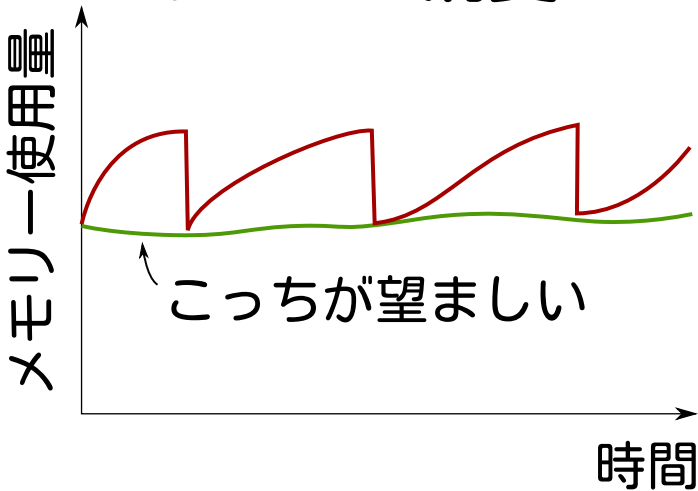
Since 2013

Groongaとmruby

- ✓ Groonga
 - ✓ 全文検索エンジン (mruby組み込み)
 - ✓ 高速に検索結果を返し続けたい
 - ✓ リソース消費は波がない方がよい
例：いらなくなったメモリーはすぐに解放

mrubyの組み込み

リソース消費



mrubyの組み込み

メモリー管理

- ✓ Groonga
 - ✓ 必要なときに確保
 - ✓ いらなくなったら解放
- ✓ mruby
 - ✓ GC
 - ✓ メモリーが足りなくなったら解放

mrubyの組み込み

GroongaとmrubyのGC

- ✓ mrubyのGCにGroongaのリソース管理を任せない
- ✓ リソース管理：
mrubyのオブジェクトのsweep時にGroongaのリソースを解放

mrubyの組み込み

mrubyのGCとリソース

- ✓ mrubyのGC
 - ✓ Groongaリソースのサイズを知らない
 - ✓ 適切なタイミングでsweepできない
 - ✓ RubyのGCも同じ

mrubyの組み込み

实例1: 明示的な解放

```
# 検索
result = table.search(condition)
begin
  output_result(result) # 出力
ensure
  result.close # 明示的な解放
end
```

mrubyの組み込み

実例2: 所有権を渡さない

```
/* Groonga側でリソース確保 */  
expr = grn_expr_create(/* ... */);  
/* mrubyのオブジェクトとしてラップ */  
mrb_expr = grn_mrb_value_from_grn_obj(mrb, expr);  
/* mruby側は参照して処理する */  
mrb_size = mrb_funcall(mrb, mrb_expr,  
                        "estimate_size", 1, mrb_table);  
/* Groonga側でリソース解放 */  
grn_expr_close(expr);
```

mrubyの組み込み

mruby組み込みのまとめ

- ✓ GCに任せないという選択
 - ✓ 使用メモリー量を安定させるため
 - ✓ 安定した性能を出すため

mrubyの組み込み

まとめ

- ✓ Rubyの組み込み
 - ✓ ガッツリ連携するつもりで設計
拡張ライブラリーで十分じゃないかよく検討すること
- ✓ mrubyの組み込み
 - ✓ アプリの大事な事を忘れないで設計
- ✓ OSS Gateもよろしく