

Rails for Ruby 1.9

moriq

Rails勉強会@関西

もくじ

- **Ruby 1.8 vs 1.9**
- **Rails for Ruby 1.9**
- **Install Ruby 1.9**
- **Benchmark**

Release schedule

- Ruby **1.8.6** – 2007/03
- Rails 1.2.3
- Ruby **1.9.1** – 2007/12
 - この時点でRailsも動くと嬉しい

Ruby **1.8** vs **1.9**

Quiz #1 String

`p "A" [0]`

`p ?A`

Quiz #1 String

p "A"[0] 65 "A"

p ?A 65 "A"

Quiz #1 String

`p "A".ord 65`

`p 65.chr "A"`

Quiz #1 String

- **String**は**bytecode**の配列
- **String**は**文字**の配列

Quiz #2 Array

```
x=[0,1];x[0,1]=nil;p x
```

```
y=[0,1].zip([2,3]);p y
```

Quiz #2 Array

```
x=[0,1];x[0,1]=nil;p x
```

```
[1]
```

```
[nil,1]
```

Quiz #2 Array

Array#[]=nil

- 要素を削除
- 値としてnilを設定

```
ENV['...'] = nil
```

```
ENV.delete('...')
```

Quiz #2 Array

```
y=[0,1].zip([2,3]);p y  
[[0,2],[1,3]]  
#<Enumerable::Enumerator:>
```

- **Stream**を効率的に扱う

Quiz #3 when

```
x=1
```

```
p case x
```

```
  when 1: "A"
```

```
  when 2: "B"
```

```
end
```

Quiz #3 when

"A"

```
syntax error, unexpected ':', expecting  
keyword_then or ',' or ';' or '¥n'  
when 1: "A"  
      ^
```

```
syntax error, unexpected keyword_when,  
expecting $end  
when 2: "B"  
      ^
```

Quiz #3 when

- コロン ':' は貴重な資源
 - Symbol :foo
 - Module Foo::Bar
 - Hash {x:1,y:2}
 - Keyword arguments
(Ruby2.0?) x:,y: = {x:1,y:2}

Quiz #3 when

```
x=1
```

```
p case x
```

```
  when 1 ; "A"
```

```
  when 2 ; "B"
```

```
end
```


Quiz #4 send

```
class C
  def one() 1 end
  private :one
end
p C.new.send(:one)
```

Quiz #4 send

1

```
in `<main>`: private method  
  `one` called for #<C:0xc285a8>  
(NoMethodError)
```

Quiz #4 send

```
class C
  def one() 1 end
  private :one
end
p C.new.funcall(:one)
p C.new.__send!(:one)
```

Quiz #4 send

	send	funcall
public	Y	Y
protected	N	N
private	N	Y

Ruby 1.8 vs 1.9

- 文法: ほぼ同じ
- メソッド: 非互換あり

Rails for Ruby 1.9

Rails for Ruby 1.9

- **Ruby 1.9対応Railsを作る**
 - **Ruby trunk (v1.9.0)**
 - **Rails 1-2 branch (v1.2.*)**
 - **ActiveSupport**
 - **ActionPack**
 - **ActiveRecord**

Dependency

- **Rails**が依存する外部ライブラリ
 - DB binding, Web server
 - Rake, RubyGems
 - cgi.rb
- **Rails**だけの対応では済まない

Unstable...

- **Ruby 1.9**はまだ不安定
 - [BUG]との戦い
- **Rails**は安定性が最重要
 - ゆっくり移行

Test First!

- テストも**1.9**対応は必要
- **Ruby**の変更に弱い

2007-03-15:

AS: 8**F** 6**E**

AP: 1**F** 0**E**

AR: 7**F** 0**E** (base_test only)

Optimizing

- **新仕様を使って効率改善**
 - **Class variable**
 - ***_methods**
 - **BasicObject**

Resources

Project site:

<http://collaboa.morIQ.com/>

SVN repos:

**[http://dev.morIQ.com/svn/
local/rails/](http://dev.morIQ.com/svn/local/rails/)**

Install Ruby 1.9

Install Ruby 1.9

- **1.8環境と共存させる**
 - **configure options**
 - **注意: #!, `ruby`**
- **RubyGemsのしくみ**
 - **\$GEM_HOME**
 - **1.9で動作[ruby-dev:30582]**

configure options

`--enable-shared`

`--prefix=~ /c`

`--program-suffix=-1.8.6`

`/home/moriq/c/bin/
ruby-1.8.6`

shebang #!

~/c/bin/gem:

```
#!/usr/bin/env ruby
```

~/c/bin/rake:

```
#!/home/moriq/c/bin/  
ruby-1.8.6
```


call ruby

`rake/Rakefile:`

```
`ruby -Ilib ./bin/rake  
  --version`
```

Install Ruby 1.9

- **1.8環境と共存させる**
 - **configure options**
 - **注意: #!, `ruby`**
- **RubyGemsのしくみ**
 - **\$GEM_HOME**
 - **1.9で動作[ruby-dev:30582]**

\$GEM_HOME

`/usr/lib/ruby/gems/1.8/...`

`export GEM_HOME=~ /c`

`/home/moriq/c/`

`cache/`

`doc/`

`gems/`

`specifications/`

Gemの構造

```
tar xvf cache/sources-0.0.1.gem
```

```
data.tar.gz
```

```
-> lib/sources.rb
```

```
metadata.gz
```

```
-> metadata (YAML)
```

- **Zlib::GzipReader**

- 1.9で動作[ruby-dev:30582]

Benchmark

Benchmark

- **benchmark.rb**
 - 注意: GC, IO
- **profiler**
 - **set_trace_func**
 - **Ruby trunk**では動かない

Rails Bench

- **gems vs vendor**

- あまり影響しない

- **ActiveSupport**

- **require**すると遅くなる

- なぜ？

ActiveRecord Bench

sqlite3-ruby select (n=1000):

- **1.8.6: 2147 req/sec**
- **1.9.0: 2320 req/sec**

with ActiveRecord:

- **1.8.6: 1100 req/sec**
- **1.9.0: 804 req/sec**

ActionPack Bench

dispatch (n=100):

- **1.8.6: 108 req/sec**
- **1.9.0: 126 req/sec**

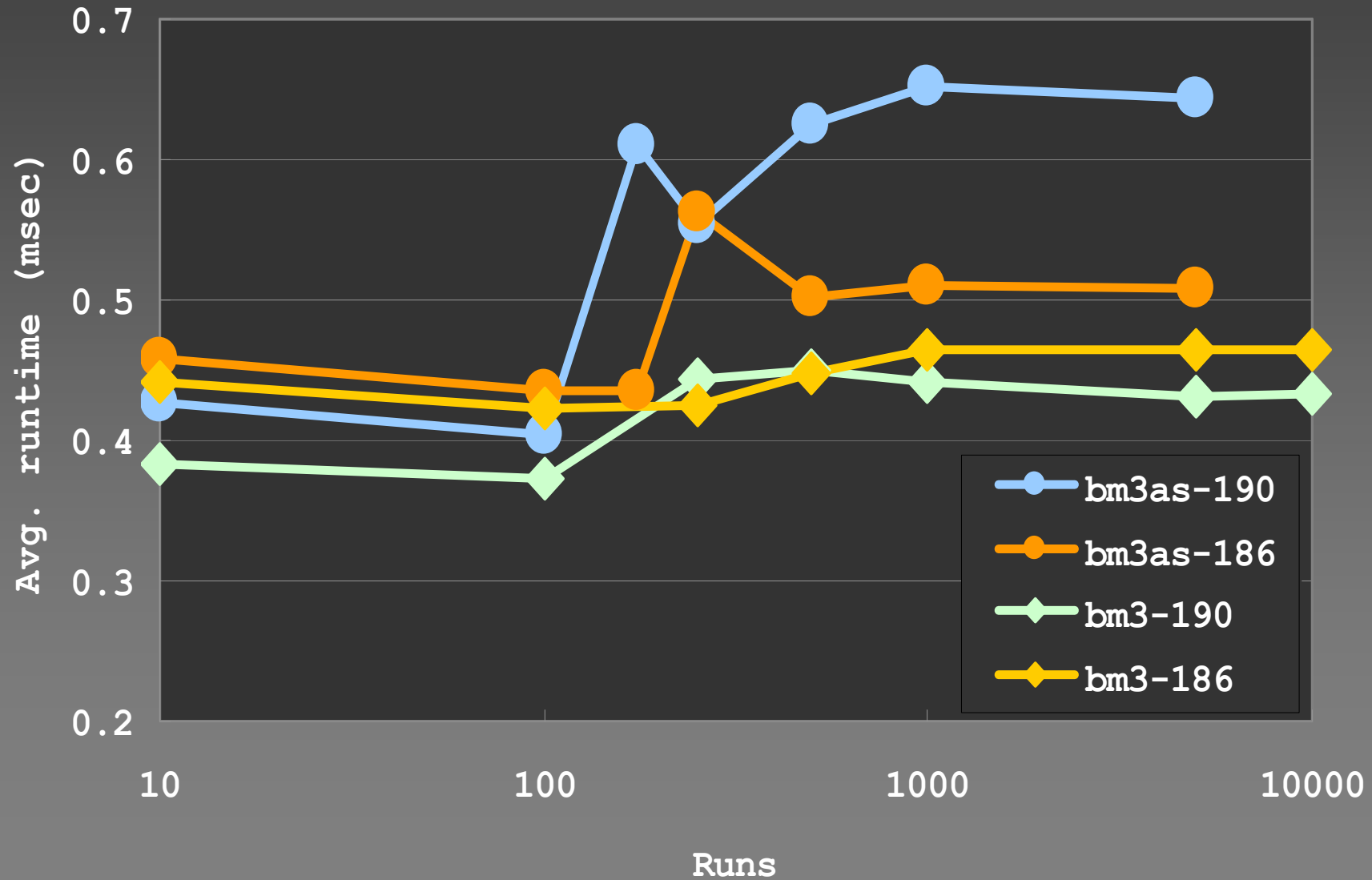
with AR (2 records):

- **1.8.6: 85.2 req/sec**
- **1.9.0: 96.2 req/sec**

Bottleneck?

- **ActiveSupport**あやしい
- **require 'active_record'**
の有無を条件に計測
- **GC**が効いている

sqlite3-ruby benchmark



まとめ

- **Rails for Ruby 1.9**は
そこそこ動作します
- **Ruby 1.9**を使ってみよう

おわり