Ruby Performance.
Tips, Tricks & Hacks

Who am I?

- Ezra Zygmuntowicz (zig-mun-tuv-itch)
- Rubyist for 4 years
- Engine Yard Founder and Architect
- Blog: http://brainspl.at

Ruby is Slow

Ruby is Slow?!?

Well, yes and no.

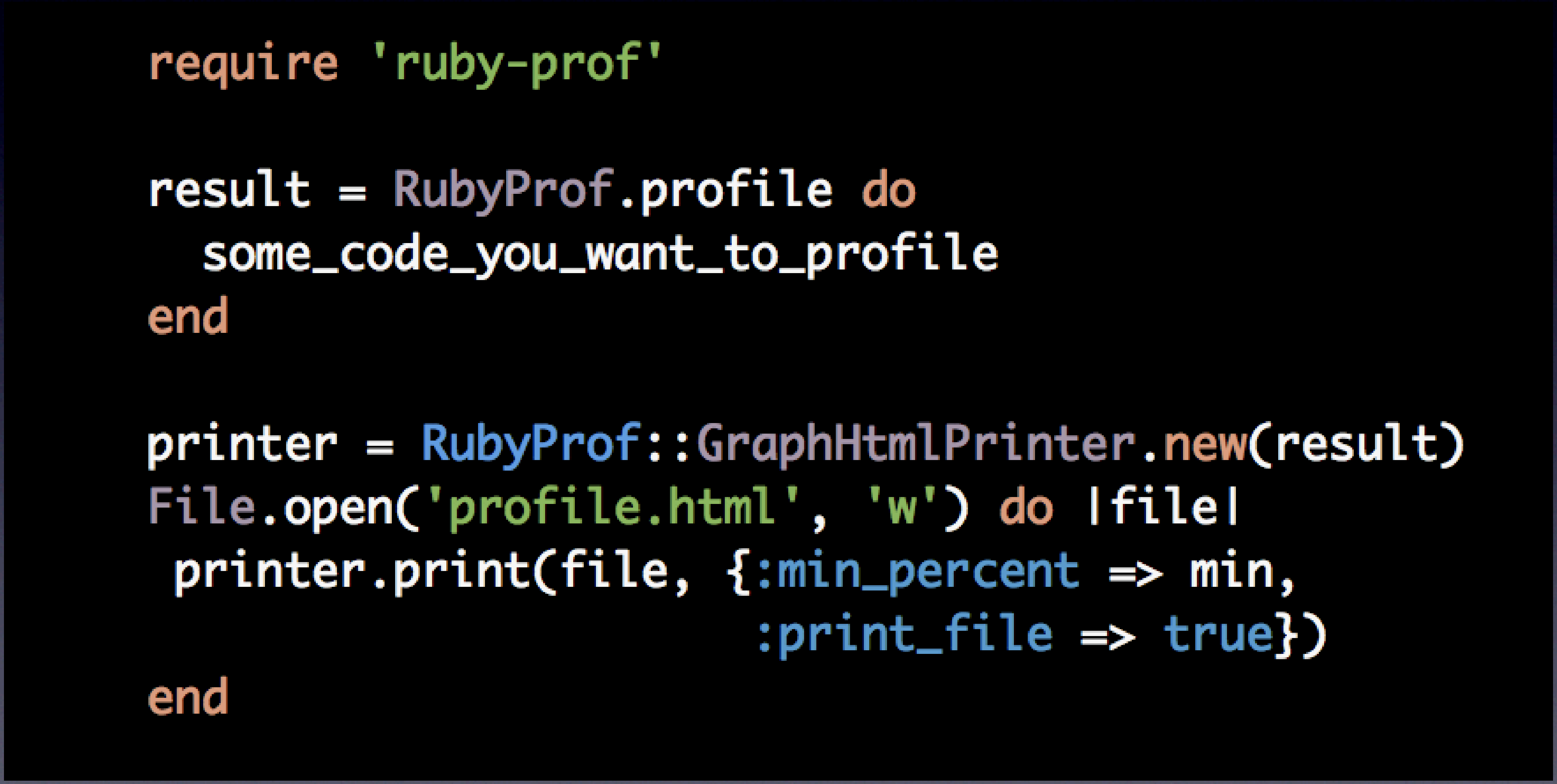
The Ruby Performance Dichotomy

Framework Code
VS
Application Code

Benchmarking: The only way to really know performance characteristics

Profiling:
Measure don’t guess.

ruby-prof



```ruby
require 'ruby-prof'

result = RubyProf.profile do
  some_code_you_want_to_profile
end

printer = RubyProf::GraphHtmlPrinter.new(result)
File.open('profile.html', 'w') do |file|
 printer.print(file, {:min_percent => min,
                       :print_file => true})
end
```

What is all this good for in real life?

Merb

Merb

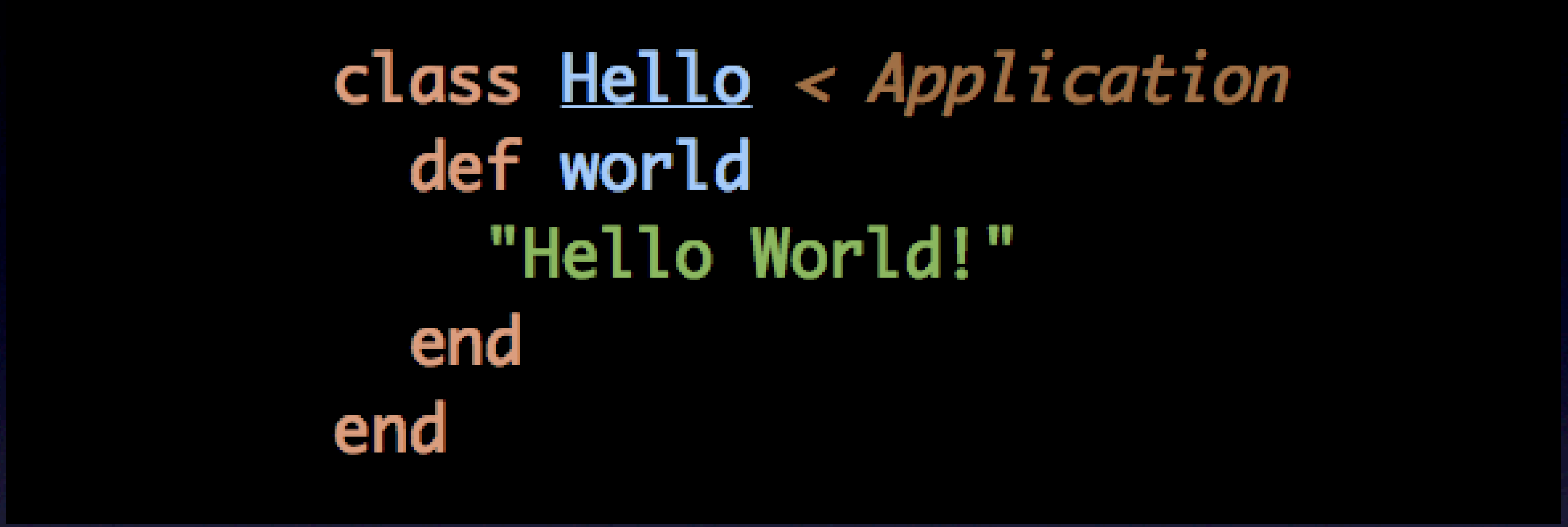
Like most useful code it started as a hack,
Merb == Mongrel + Erb

- No cgi.rb !!
- Clean room implementation of ActionPack
- Thread Safe with configurable Mutex Locks
- Rails compatible REST routing
- No Magic(well less anyway ;)
- Did I mention no cgi.rb?
- Fast! On average 2-4 times faster than rails

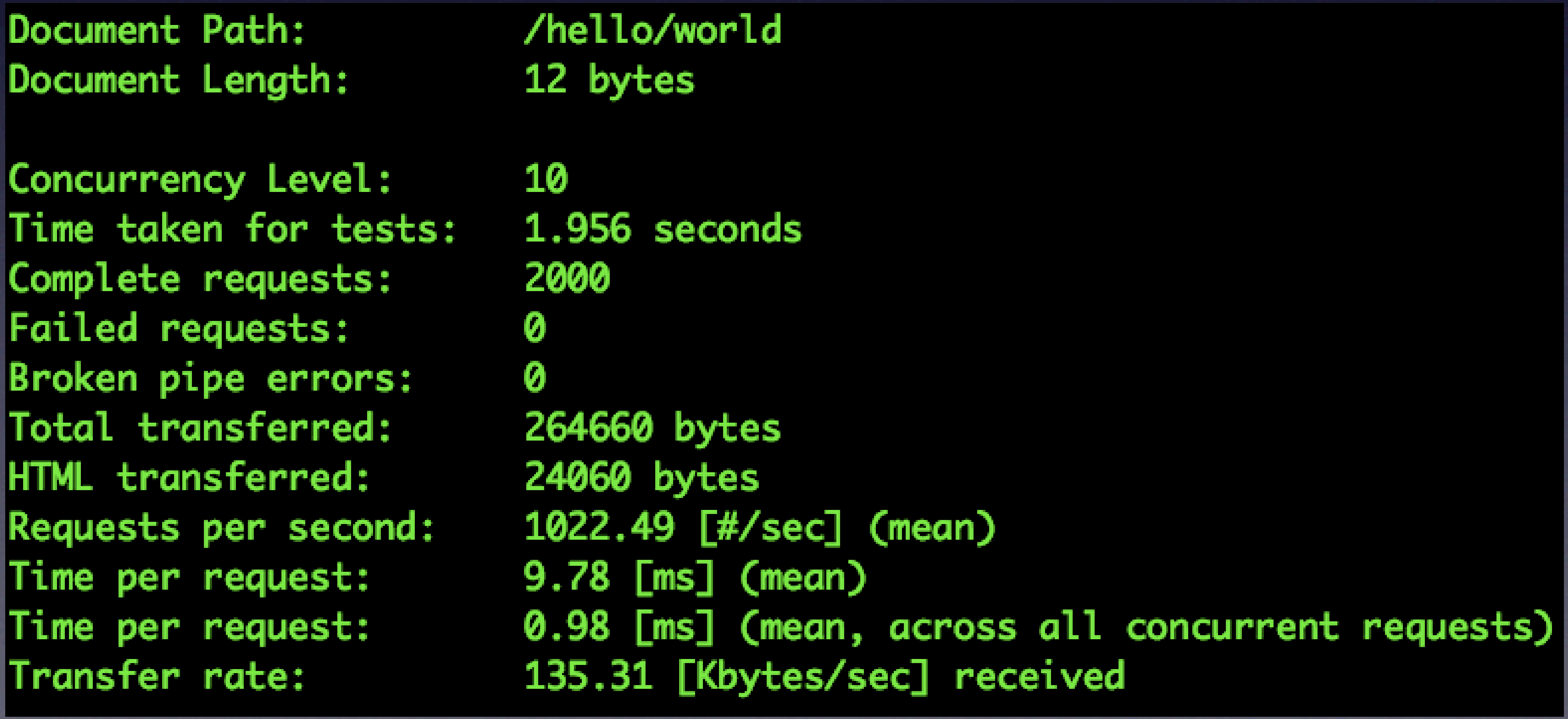
Design Goals

- Small core framework for the VC in MVC
- ORM agnostic, use ActiveRecord, Sequel, DataMapper or roll your own db access.
- Prefer simple code over magic code
- Keep the stack traces short(I’m looking at you alias_method_chain)
- Thread safe, reentrant code

Merb Hello World



```
class Hello < Application
  def world
    "Hello World!"
  end
end
```



```
Document Path:          /hello/world
Document Length:        12 bytes

Concurrency Level:      10
Time taken for tests:   1.956 seconds
Complete requests:      2000
Failed requests:        0
Broken pipe errors:     0
Total transferred:      264660 bytes
HTML transferred:       24060 bytes
Requests per second:    1022.49 [#/sec] (mean)
Time per request:       9.78 [ms] (mean)
Time per request:       0.98 [ms] (mean, across all concurrent requests)
Transfer rate:          135.31 [Kbytes/sec] received
```

No code is faster then no code

- Simplicity and clarity trumps magic every time.
- When in doubt leave it out.
- Core framework to stay small and simple and easy to extend without gross hacks
- Prefer plugins for non core functionality
- Plugins can be gems

Key Differences

- No auto-render. The return value of your controller actions is what gets returned to client
- Merb’s render method just returns a string, allowing for multiple renders and more flexibility
- PartController’s allow for encapsualted applets without big performance cost

Why not work on Rails instead of making a new framework?

- Originally I was trying to optimize Rails and make it more thread safe. In the end it was easier to make a new framework from scratch then to try and change ActionPack
- There is too much backwards compatibility baggage in the rails code base to make radical changes without breaking people apps and pissing them off
- I love Rails but it does punish you if you go off the ‘Golden Path’
- Merb makes fewer assumptions on how it will be used, you create your own ‘Golden Path’.
- Merb is a hackers framework.You can built it up to your problem domain since the core is easily extendable.
- Kitchen sink not included...

Current Status

- Quite a few merb apps in production
- Pushing towards 0.4 release in the next few weeks.
- Once 0.4 is out the http://merbivore.com website will be launched
- 0.4 will most likely be feature complete for core framework.
- This means the core will stay light, any additional features will be plugins from the get go

What is in Merb's future?

- Rubinius compatibility
- More profiling and optimizations
- More tools and creature comforts
- Core will stay lean, functionality added by plugins

What still needs work?

- Doc patches and more specs always welcome. Probably the easiest way to dive in
- Spec harness for merb applications still needs some work. If your a spec guru then we'd love to have your help
- Refactoring the framework/app/plugin initialization and configuration system.
- Write tutorials or blog articles as you learn merb

How can you get involved?

- Merb has an open door svn policy
- Send in a few good patches and you will get commit rights to the repo
- Merb irc channel is active, #merb on irc.freenode.net
- Mailing list: http://rubyforge.org/mailman/listinfo/merb-devel
- Trac: http://merb.devjavu.com/

Questions?

Rubinius Demo (if we have time)