# Ruby profiling

David Grayson

Las Vegas Ruby Meetup

2013-08-28


SLOW
RUBY
RUNNING

# Two profiling tools

- ruby-prof
- rblineprof

# Features of ruby-prof

- Speed
- Can measure:
  - call times
  - memory usage
  - object allocations
- Text and HTML reports:
  - Flat profiles
  - Graph profiles
  - Call tree profiles for KCacheGrind
- Supports multiple threads

# Example ruby-prof profiles

# ruby-prof executable

```
ruby-prof slow_program.rb
```

```
ruby-prof --file=profile.html \
          --printer=graph_html \
          slow_program.rb
```

# ruby-prof API

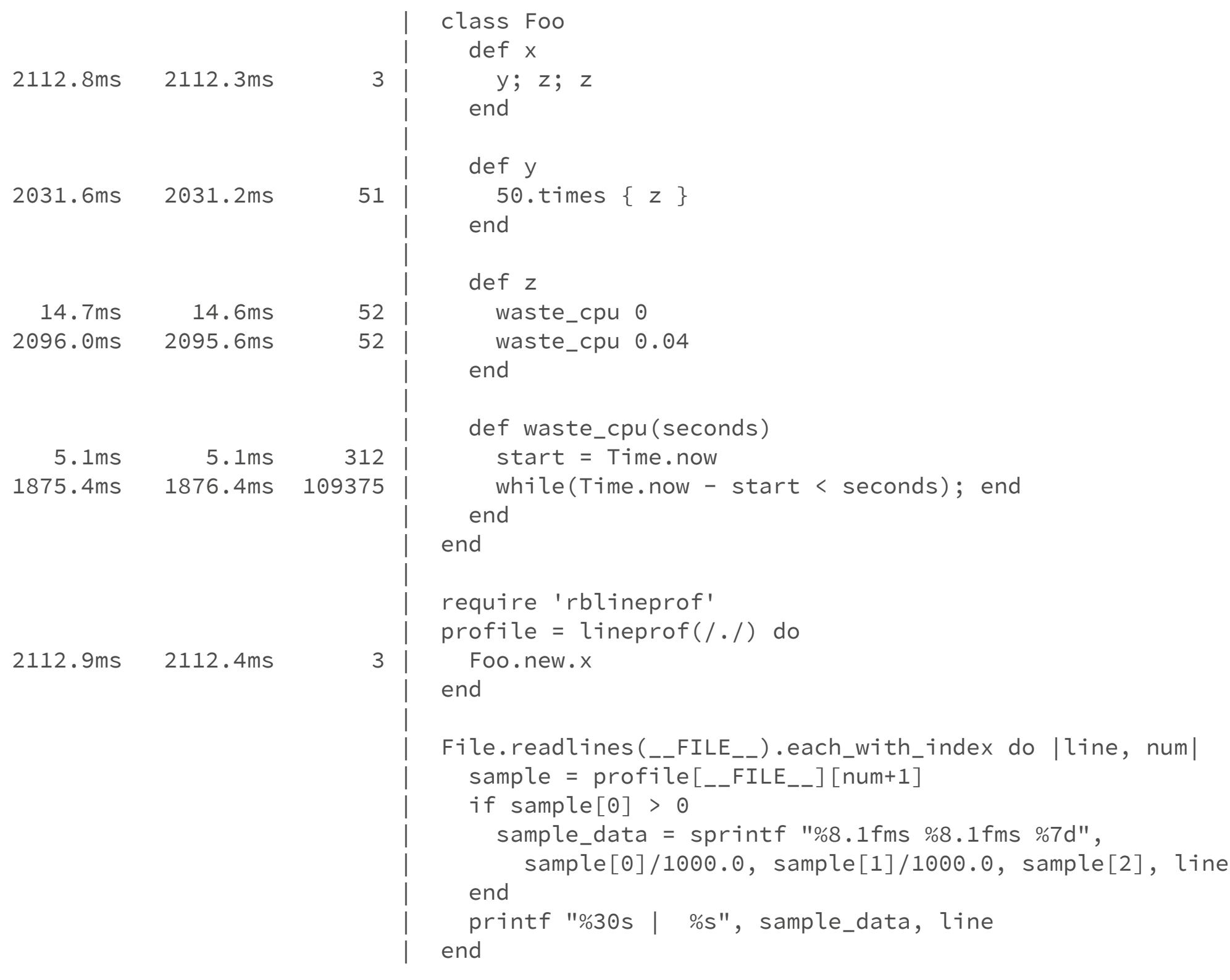
```
require 'ruby-prof'
RubyProf.start

slow_code

result = RubyProf.stop
File.open("slow2.txt","w") do |file|
  RubyProf::GraphPrinter.new(result).print(file)
end
```

# Excluding methods in ruby-prof


- Exclude methods by class/method name or regular expression.
- Remove useless things like `Integer#times` from reports.

# rblineprof

- 652-line ruby extension written in C
- Seems to report the time spent on each line.
- No documentation, no report formatting


class Foo
def x
y; z; z
end
def y
50.times { z }
end
def z
waste_cpu 0
waste_cpu 0.04
end
def waste_cpu(seconds)
start = Time.now
while(Time.now - start < seconds); end
end
end
require 'rblineprof'
profile = lineprof(/./) do
Foo.new.x
end
File.readlines(__FILE__).each_with_index do |line, num|
sample = profile[__FILE__][num+1]
if sample[0] > 0
sample_data = sprintf "%8.1fms %8.1fms %7d",
sample[0]/1000.0, sample[1]/1000.0, sample[2], line
end
printf "%30s | %s", sample_data, line
end
2112.8ms 2112.3ms 3
2031.6ms 2031.2ms 51
14.7ms 14.6ms 52
2096.0ms 2095.6ms 52
5.1ms 5.1ms 312
1875.4ms 1876.4ms 109375
2112.9ms 2112.4ms 3

# References

- ruby-prof
  - https://github.com/ruby-prof/ruby-prof/
- rblineprof
  - https://github.com/tmm1/rblineprof
  - https://github.com/blog/1475-escape-velocity