# Scanning Strings at Supersonic Speed

# @

# murphy

Everything
that can go wrong,
will go wr

## (Kornelius Kalnbach)

## murphy@rubychan.de

## @murphy_karasu

## murfy.de

## sofatutor.com

# Scanning

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est lakorum.

# Strings

"Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est lakorum."

# Strings

"<!DOCTYPE html><html lang="en"><head><meta
charset="utf-8"><title>Agenda –EuRuKo 2011</
title><link rel="stylesheet" href="/stylesheets/
screen.css"><link rel="stylesheet" href="fancybox/
jquery.fancybox-1.3.4.css"><linkrel="alternate"
type="application/atom+xml" title="ATOM
feed"href="http://euruko2011.org/feed.atom"><link
rel="alternate"type="application/atom+xml"
title="github feed"…"

# Strings

```
"class Song
  def initialize name, author
    @name = name
    @author = author
  end
  def to_s
    "this is :#{@name}----#{@author}*****"
  end
end"
```

# Strings

```
"class Song
  def initialize name, author
    @name = name
    @author = author
  end
  def to_s
    "this is :#{@name}----#{@author}*****"
  end
end".scan
```

# Supersonic

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 3KB | | | | | | | | |

Mach 1.8 = 1900 km/h

2514 pages per second

7.5 MB/s

# Speed

Why?

- Syntax Highlighting
- Parsing (eg. HAML)
- Rite

# Shootout


12.283
Syntax
11.571
Pygments
3.565
source-
highlight
1.811
3.396
CodeRay

# Target

write a supersonic scanner
with pure Ruby code

# Resources

a fast machine

# Resources

big examples

5+ seconds JRuby

here: 160 MB

= 9 times ruby-head

# Resources

rvm ♥ ♥ ♥

# Resources

Time

Endurance

Craziness

# General Ideas

C

- avoid convenient APIs
- write everything yourself
- write for the CPU

# General Ideas

Ruby

- embrace the core libraries
- write less code
- write for the interpreter

# General Ideas

Ruby

- embrace the core libraries
- write less code
- write for the interpreter

# String

```
s = "<head><title>EuRuKo 2011</title>"
```

# RegExp

```
s = "<head><title>EuRuKo 2011</title>"

pattern = /<(\w+)/
```

# String + RegExp

```ruby
s = "<head><title>EuRuKo 2011</title>"

puts s[/<(\w+)/, 1]
```

```
head
```

# String + RegExp

```ruby
s = "<head><title>EuRuKo 2011</title>"

puts s.scan(/<(\w+)/)
```

```
head
title
```

# String + RegExp

```
s = "<head><title>EuRuKo 2011</title>"

s.scan(/<(\w+)/) do |tag|
  puts tag
end
```

```
head
title
```

# StringScanner

```
scanner = StringScanner.new(s)

until scanner.eos?
  if scanner.scan(/<(\w+)/)
    tag = scanner[1]
  else
    scanner.getch
  end
end
```

# StringScanner

Why?

- avoid big RegExp
- control the scan process
- use patterns depending on state
- create patterns on the fly

# Benchmark

```
s = "<head><title>EuRuKo 2011</title>"

s *= 5_000_000

s.scan(/<(\w+)/) do |tag|
  tag
end
```

| | |
|---|---|
| sonic™ | 21.2 |
| 1.8.7 | 13.6 |
| 1.9.2 | 5.4 |
| jruby | 4.7 |
| rbx | 29.1 |
| mac | 34.7 |

# StringScanner

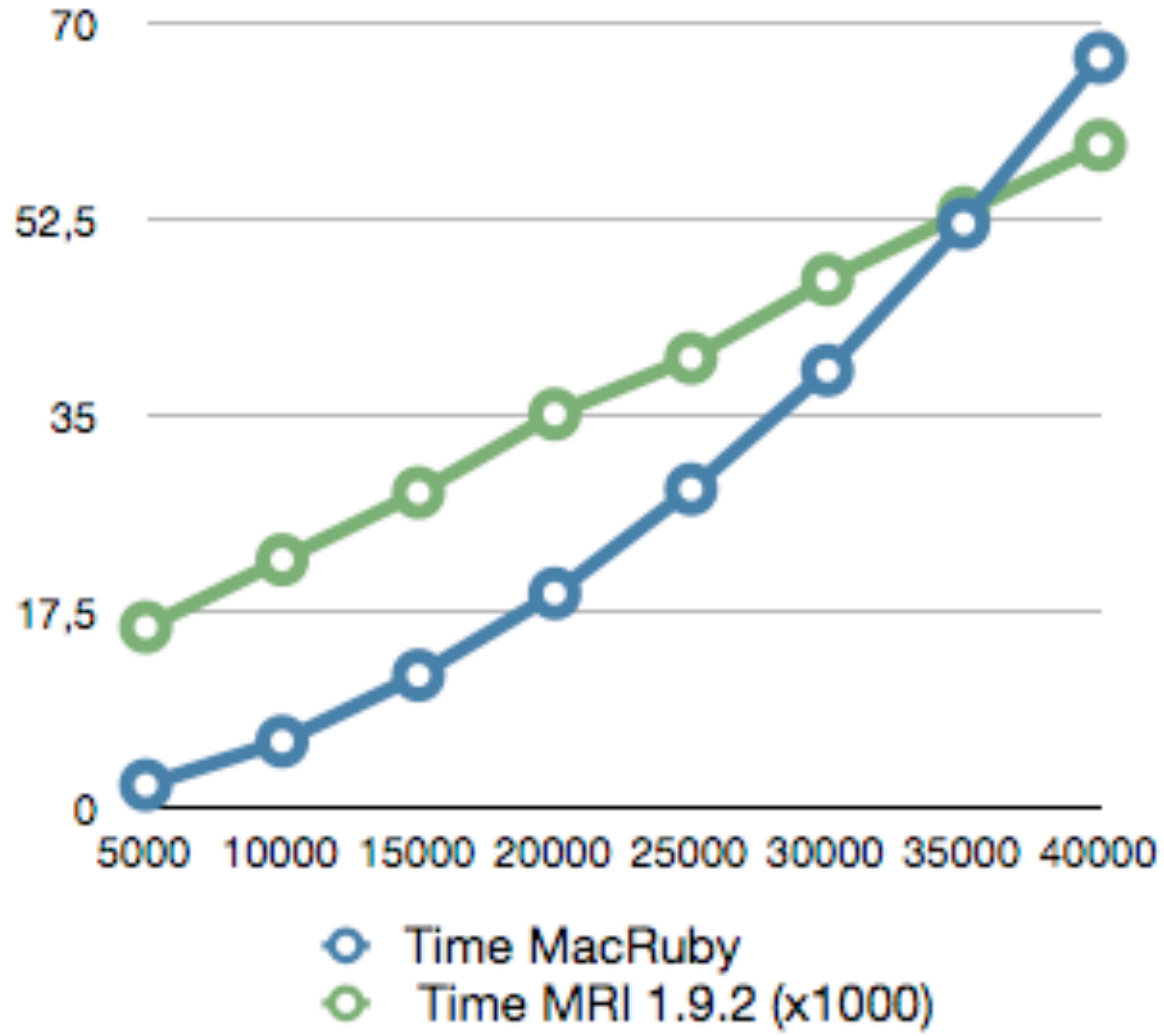
```
scanner = StringScanner.new(s)

until scanner.eos?
  if scanner.scan(/<(\w+)/)
    tag = scanner[1]
  else
    scanner.getch
  end
end
```

| | |
|---|---|
| sonic™ | 21.2 |
| 1.8.7 | 61.0 |
| 1.9.2 | 40.0 |
| jruby | 21.1 |
| rbx | 180.1 |
| mac | ? |

# StringScanner on MacRuby

http://www.macruby.org/trac/ticket/938

| String length | Time MacRuby | Time MRI 1.9.2 (times 1000) |
|---|---|---|
| 5000 | 1,9 | 16 |
| 10000 | 5,8 | 22 |
| 15000 | 11,7 | 28 |
| 20000 | 19,0 | 35 |
| 25000 | 28,3 | 40 |
| 30000 | 38,9 | 47 |
| 35000 | 52,0 | 53 |
| 40000 | 66,8 | 59 |



# StringScanner

```
scanner = StringScanner.new(s)

until scanner.eos?
  if scanner.scan(/<(\w+)/)
    tag = scanner[1]
  else
    scanner.getch
  end
end
```

| sonic™ | 21.2 |
|---|---|
| 1.8.7 | 13.6 |
| 1.9.2 | 5.4 |
| jruby | 4.7 |
| rbx | 29.1 |
| mac | 34.7 |

| sonic™ | 21.2 |
|---|---|
| 1.8.7 | 61.0 |
| 1.9.2 | 40.0 |
| jruby | 21.1 |
| rbx | 180.1 |
| mac | ? |

# General Ideas

Ruby

- embrace the core libraries
- write less code
- write for the interpreter

# Less Code

```
class HTML
  def initialize html
    @scanner = StringScanner.new(html)
  end

  def scan
    until @scanner.eos?
      if @scanner.scan(/<(\w+)/)
        yield @scanner[1]
      ...
  end
end
```

# Less Code

```
class HTML
  def initialize html
    @scanner = StringScanner.new(html)
  end

  def scan
    until eos?
      if scan(/<(\w+)/)
        yield self[1]
      ...
  end
end
```

# Less Code

```
class HTML
  def initialize html
    @scanner = StringScanner.new(html)
  end
  delegate :eos?, :scan, :[]
  def scan
    until eos?
      if scan(/<(\w+)/)
        yield self[1]
      ...
  end
end
```

# Less Code

```
class HTML
  def initialize html
    @scanner = StringScanner.new(html)
  end
  delegate :eos?, :scan, :[]
  def tokenize
    until eos?
      if scan(/<(\w+)/)
        yield self[1]
      ...
  end
end
```

# Less Code

```
class HTML < StringScanner
  def initialize html
    super
  end

  def tokenize
    until eos?
      if scan(/<(\w+)/)
        yield self[1]
      ...
  end
end
```

# Less Code

```
class HTML < StringScanner
  def tokenize
    until eos?
      if scan(/<(\w+)/)
        yield self[1]
      ...
  end
end
```

# General Ideas

Ruby

- embrace the core libraries
- write less code
- write for the interpreter

```
out = Encoder.new.encode Scanner.new(in)
```

- Scanner: simple, 9 rules
- Encoder: does nothing

# Single Core

```
out = Encoder.new.encode Scanner.new(in)
```

- Scanner: simple, 9 rules
- Encoder: does nothing

| | |
|---|---|
| sonic™ | 21.2 |
| jruby | 42.3 |
| 1.9 | 72.4 |

# Multicore

```
threads = []

in.lines.each_slice 300_000 do |lines|
  threads << Thread.new do
    chunk = lines.join
    Thread.current[:out] = Encoder.new.encode Scanner.new chunk
  end
end

threads.each(&:join)

out = threads.map { |thread| thread[:out] }.join
```

| | |
|---|---|
| sonic™ | 21.2 |
| jruby | 26.5 |
| 1.9 | 74.4 |

# Multicore

```ruby
threads = []
chunk_offsets = [0]
in.lines.each_slice slice_size do |lines|
  chunk_offsets << chunk_offsets.last + lines.join.bytesize
end

chunk_offsets.each_cons(2) do |this_chunk, next_chunk|
  threads << Thread.new do
    chunk = code[this_chunk...next_chunk]
    Thread.current[:out] = Encoder.new.encode Scanner.new chunk
  end
end

threads.each(&:join)

out = threads.map { |thread| thread[:out] }.join
```

| | |
|---|---|
| sonic™ | 21.2 |
| jruby | 24.2 |
| 1.9 | 73.6 |

# Multicore

```
threads = []
chunk_offsets = [0]
in.lines.each_slice slice_size do |lines|
  chunk_offsets << chunk_offsets.last + lines.join.bytesize
end

chunk_offsets.each_cons(2) do |this_chunk, next_chunk|
  threads << Thread.new do
    chunk = code[this_chunk...next_chunk]
    Thread.current[:out] = Encoder.new.encode Scanner.new chunk
  end
end

threads.each(&:join)

out = threads.map { |thread| thread[:out] }.join
```

| | |
|---|---|
| input | 160 MB |
| join | 793 MB |
| offset | 440 MB |

# MORE MULTICORE

```
threads = []
chunk_offsets = [0]
in.lines.each_slice slice_size do |lines|
  chunk_offsets << chunk_offsets.last + lines.join.bytesize
end

chunk_offsets.each_cons(2) do |this_chunk, next_chunk|
  threads << Thread.new do
    chunk = code[this_chunk...next_chunk]
    Thread.current[:out] = Encoder.new.encode Scanner.new chunk
  end
end

threads.each(&:join)

out = threads.map { |thread| thread[:out] }.join
```

| | |
|---|---|
| sonic™ | 21.2 |
| 2 cores | 24.2 |
| 4 cores | 17.6 |

# Questions?

not allowed:

- When will CodeRay 1.0 be released?

# Thank you!

- @euruko
- @yukihiro_matz
- @bovensiepen
- @heinz_gies
- my girlfriend