

# Rails Performance

Michael Koziarski  
[michael@koziarski.com](mailto:michael@koziarski.com)



# Rails Performance

Relax

Programmers Love  
Optimisation

Science Based

Objective

Provable

# Opportunity Cost



Is this optimisation  
**really** the best use of  
your time?

# Two Justifications

# Hardware Costs

Responsiveness

# Hardware Costs

**Profits = Revenue - Costs**

Hardware isn't all of  
your costs

Hardware costs are not  
continuous



Hardware costs are not  
continuous

Unless you're **huge**

Hardware costs are not  
continuous

Unless you're **huge**

You're **not** huge

Hardware is basically  
free

Hardware is basically  
free

Compared to staff

# Hardware Cost Driven Optimisation

# Example.com

- 4 Servers at \$299/mo
- Total hosting of ~ \$1,200/mo

# Example.com

- 10% performance improvement across the board!
- Leads to a \$120/mo saving...

# Example.com

- Staff costing 60-80k/yr
- \$28-\$38/hr



**It almost certainly  
wasn't worth it**

Responsiveness

**Much more Important**

Slow applications are  
no fun to use

No Users

No Revenue

**No Business**

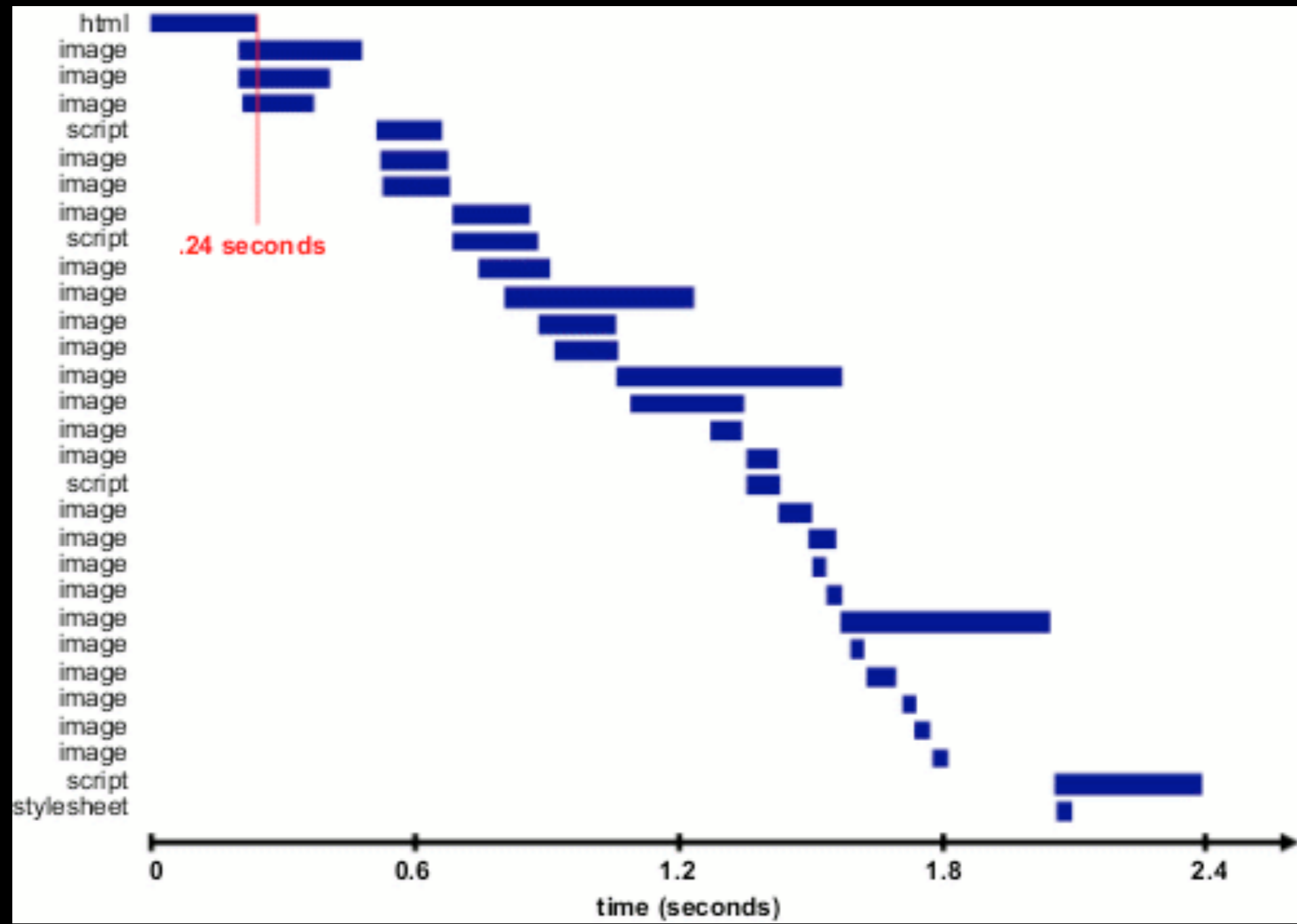




Not about requests per  
second

Seconds per request

# Perceived Performance



Firebug - Facebook   Michael Koziarski									
Inspect	Clear	All	HTML	CSS	JS	XHR	Images	Flash	Search
Console	HTML	CSS	Script	DOM	Net	YSlow	Options		
▶ GET profile.php?id=	200 OK				new.facebook.com	42 KB			853ms
▶ GET upload.css	200 OK				static.ak.fbcdn.net	395 B			364ms
▶ GET picture.css	200 OK				static.ak.fbcdn.net	871 B			406ms
▶ GET app_source.css	200 OK				static.ak.fbcdn.net	351 B			410ms
▶ GET upload.css	200 OK				static.ak.fbcdn.net	2 KB			393ms
▶ GET pillfilter.css	200 OK				static.ak.fbcdn.net	581 B			388ms
▶ GET feedcrawl.css	200 OK				static.ak.fbcdn.net	2 KB			404ms
▶ GET editprofile.js	200 OK				static.ak.fbcdn.net	2 KB			96ms
▶ GET owner.js	200 OK				static.ak.fbcdn.net	7 KB			95ms
▶ GET upload.js	200 OK				static.ak.fbcdn.net	712 B			95ms
▶ GET square.js	200 OK				static.ak.fbcdn.net	2 KB			109ms
▶ GET remove.js	200 OK				static.ak.fbcdn.net	617 B			94ms
▶ GET token.js.pkg.ph	200 OK				static.ak.fbcdn.net	4 KB			134ms
▶ GET untrusted.js	200 OK				static.ak.fbcdn.net	558 B			97ms
▶ GET custom_source,	200 OK				static.ak.fbcdn.net	559 B			102ms
▶ GET apps.js	200 OK				static.ak.fbcdn.net	753 B			96ms
▶ GET camera.js	200 OK				static.ak.fbcdn.net	697 B			104ms
▶ GET progress_bar.js	200 OK				static.ak.fbcdn.net	953 B			98ms
▶ GET upload.js	200 OK				static.ak.fbcdn.net	3 KB			106ms
▶ GET feedcrawl.js	200 OK				static.ak.fbcdn.net	2 KB			102ms
▶ GET editpencil.gif	200 OK				static.ak.fbcdn.net	148 B			105ms
▶ GET q_silhouette.gif	200 OK				static.ak.fbcdn.net	390 B			167ms
▶ GET 236.png?2:4569	200 OK				static.ak.fbcdn.net	3 KB			193ms
▶ GET 369.png?2:9043	200 OK				static.ak.fbcdn.net	2 KB			198ms
▶ GET 212.png?2:4569	200 OK				static.ak.fbcdn.net	3 KB			190ms
▶ GET 72.png?2:45699	200 OK				static.ak.fbcdn.net	6 KB			252ms
▶ GET profile_new_tab	200 OK				static.ak.fbcdn.net	130 B			207ms
▶ GET q537695903_90	200 OK				profile.ak.facebook.com	3 KB			871ms
▶ GET composer_tab_s	200 OK				static.ak.fbcdn.net	732 B			177ms
▶ GET composer_tab_s	200 OK				static.ak.fbcdn.net	83 B			196ms
▶ GET friend_guy.gif?2	200 OK				static.ak.fbcdn.net	350 B			184ms
▶ GET post.gif?2:2657	200 OK				static.ak.fbcdn.net	222 B			193ms
▶ GET note.gif?2:2579	200 OK				static.ak.fbcdn.net	159 B			190ms
▶ GET q706645624_1!	200 OK				profile.ak.facebook.com	3 KB			755ms
▶ GET q687120081_6:	200 OK				profile.ak.facebook.com	3 KB			969ms
▶ GET n689275123_1:	200 OK				profile.ak.facebook.com	9 KB			1.2s

Firebug - Facebook   Michael Koziarski							Options
Inspect Clear All HTML CSS JS XHR Images Flash							Q
Console	HTML	CSS	Script	DOM	Net	YSlow	
▶	GET	video_play_ovei	200 OK	static.ak.fbcdn.net	505 B		2.2s
▶	GET	start_quote_gra	200 OK	static.ak.fbcdn.net	129 B		2.21s
▶	GET	video_play_ovei	200 OK	static.ak.fbcdn.net	530 B		2.22s
▶	GET	end_quote_gray	200 OK	static.ak.fbcdn.net	131 B		2.25s
▶	GET	string_image.ph	200 OK	new.facebook.com	?		2.16s
▶	GET	app_2_2137013	200 OK	photos-a.ak.facebook.com	310 B		2.73s
▶	GET	app_2_1047335	200 OK	photos-c.ak.facebook.com	246 B		2.7s
▶	GET	app_2_1455058	200 OK	photos-c.ak.facebook.com	1008 B		2.53s
▶	GET	app_2_1368610	200 OK	photos-c.ak.facebook.com	218 B		2.52s
▶	GET	app_2_1740443	200 OK	photos-c.ak.facebook.com	626 B		2.51s
▶	GET	app_2_1053858	200 OK	photos-g.ak.facebook.com	89 B		2.6s
▶	GET	app_2_8032453	200 OK	photos-949.ll.facebook.com	353 B		2.68s
▶	GET	app_2_3206124	200 OK	photos-f.ak.facebook.com	609 B		2.56s
▶	GET	t555278697_14	200 OK	photos-c.ak.facebook.com	3 KB		2.44s
▶	GET	app_2_8584254	200 OK	photos-e.ak.facebook.com	323 B		2.58s
▶	GET	app_2_1256525	200 OK	photos-e.ak.facebook.com	538 B		2.56s
▶	GET	app_2_1215659	200 OK	photos-665.ll.facebook.com	399 B		2.58s
▶	GET	app_2_1143505	200 OK	photos-d.ak.facebook.com	373 B		2.57s
▶	GET	t555278697_14	200 OK	photos-f.ak.facebook.com	4 KB		2.48s
▶	GET	t555278697_14	200 OK	photos-e.ak.facebook.com	2 KB		2.48s
▶	GET	t555278697_14	200 OK	photos-e.ak.facebook.com	3 KB		2.51s
▶	GET	app_1_2231777	200 OK	photos-h.ak.facebook.com	6 KB		2.52s
▶	GET	safe_image.php	200 OK	external.ak.fbcdn.net	4 KB		2.5s
▶	GET	safe_image.php	200 OK	external.ak.fbcdn.net	3 KB		358ms
▶	GET	safe_image.php	200 OK	external.ak.fbcdn.net	2 KB		162ms
▶	GET	safe_image.php	200 OK	external.ak.fbcdn.net	2 KB		106ms
▶	GET	safe_image.php	200 OK	external.ak.fbcdn.net	6 KB		136ms
▶	GET	safe_image.php	200 OK	external.ak.fbcdn.net	2 KB		621ms
▶	GET	token.gif	200 OK	static.ak.fbcdn.net	2 KB		969ms
▶	GET	token_selected.	200 OK	static.ak.fbcdn.net	2 KB		985ms
▶	GET	token_x.gif	200 OK	static.ak.fbcdn.net	64 B		984ms
▶	GET	token_hover.gif	200 OK	static.ak.fbcdn.net	2 KB		1.01s
▶	GET	p_689275123=	200 OK	0.channel15.facebook.com	25 B		55.63s
▶	GET	p_689275123=	200 OK	0.channel15.facebook.com	25 B		55.65s
▶	GET	p_689275123=	200 OK	0.channel15.facebook.com	25 B		55.62s

77 requests

154 KB

2.98m

853ms

3:00



PHP is 0.4% of the total

Assume it gets 50%  
faster

The net saving is 0.2%

Not Noticeable

Frontend Performance  
should be your Priority

YSlow

Inspect Performance Stats Components Tools Help  
Console HTML CSS Script DOM Net YSlow

Performance Grade: A (92)

- C 1. Make fewer HTTP requests  
This page has 6 external JavaScript files.  
This page has 11 CSS background images.
- A 2. Use a CDN
- B 3. Add an Expires header
- A 4. Gzip components

★★★★★  
[12 reviews](#)

40,489 weekly downloads

564,129 total downloads



## YSlow 0.9.5b2

by Yahoo! Inc.

Web Development

YSlow analyzes web pages and tells you why they're slow based on Yahoo's rules for high performance web sites.

Updated June 17, 2008

Download Now



### Performance Grade: D (62)

<b>B</b>	1. Make fewer HTTP requests ▶
<b>F</b>	2. Use a CDN ▶
<b>F</b>	3. Add an Expires header ▶
<b>C</b>	4. Gzip components ▶
<b>A</b>	5. Put CSS at the top
<b>A</b>	6. Put JS at the bottom
<b>A</b>	7. Avoid CSS expressions
<b>n/a</b>	8. Make JS and CSS external ▶
<b>C</b>	9. Reduce DNS lookups ▶
<b>A</b>	10. Minify JS ▶
<b>A</b>	11. Avoid redirects
<b>A</b>	12. Remove duplicate scripts
<b>F</b>	13. Configure ETags ▶



**B** 1. Make fewer HTTP requests ▾

**This page has 5 external JavaScript files.  
This page has 8 CSS background images.**

#### C 4. Gzip components ▾

**These components are not gzipped:**

 **(6.1K) <http://www.koziarski.com/>**

 **(5.9K) <http://www.koziarski.com/stylesheets/styleSheet.css>**

Fix all of these

**I've done all that!**

# Optimisation Procedure

# Optimisation Procedure

1. Figure out what's slow

# Optimisation Procedure

1. Figure out what's slow
2. Figure out why it's slow

# Optimisation Procedure

1. Figure out what's slow
2. Figure out why it's slow
3. Make it faster



# **I. Find What's Slow**

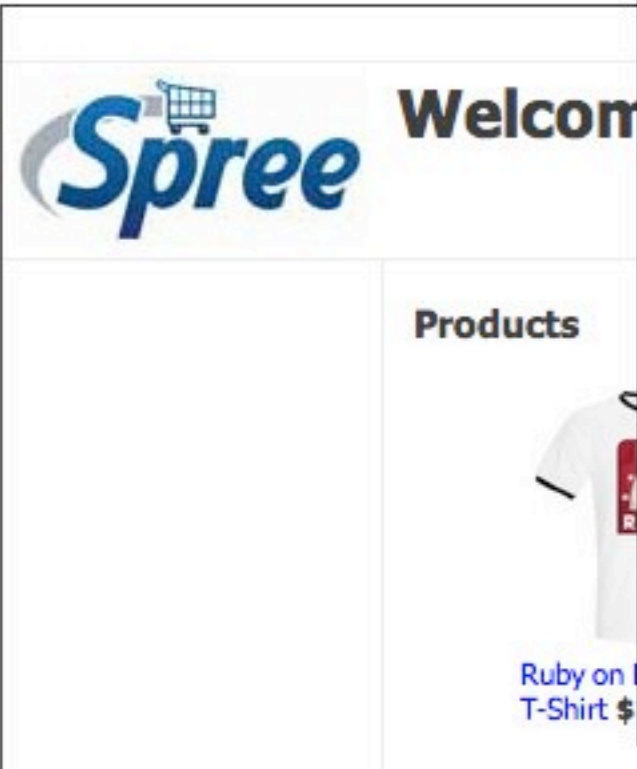
Choose your target

# Choose your target

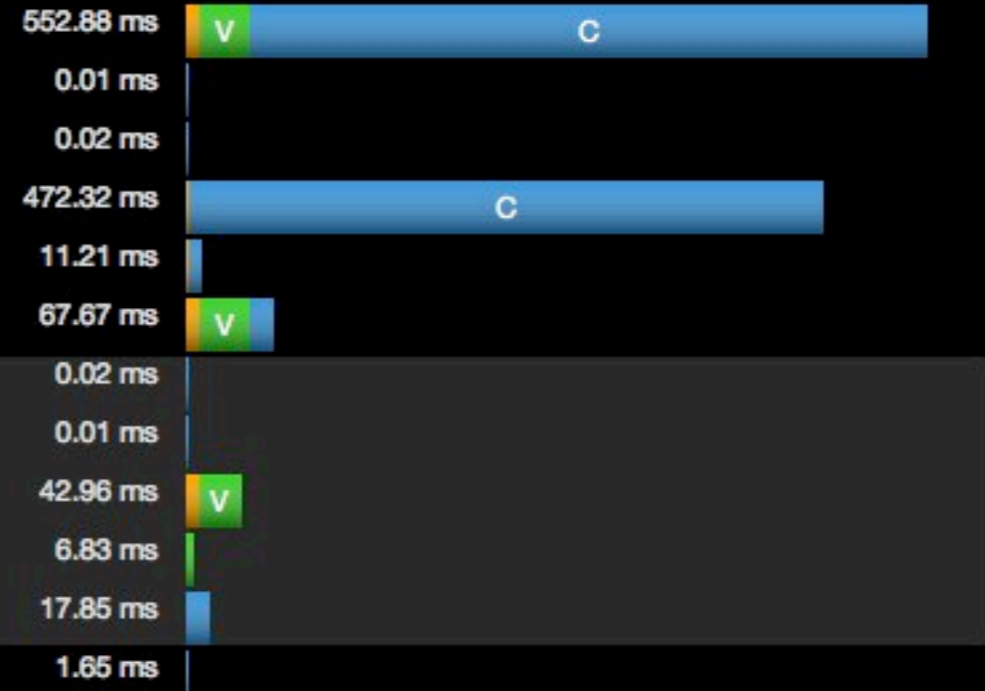
- Slower than the rest

# Choose your target

- Slower than the rest
- Heavily used

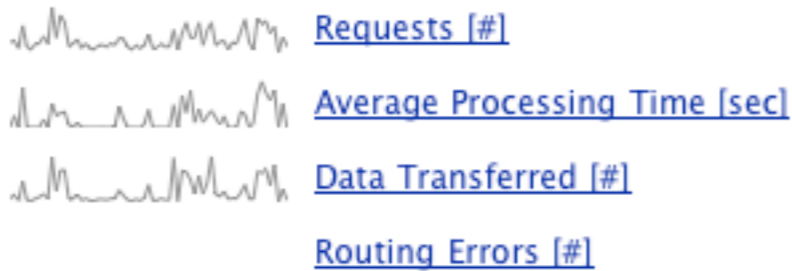


- ▼ Perform **Index** action in **ProductsController**
  - Before filter **instant\_controller\_and\_action\_names**
  - Before filter **verify\_authenticity\_token**
  - ▶ Before filter **set\_locale**
  - ▶ Before filter **find\_cart**
  - ▼ Invoke **Index** action
    - Invoke **belongs\_to** action
    - Invoke **object\_name** action
    - ▶ Render file **products/index**
    - ▶ Render file **layouts/application**
    - (Other)
    - (Other)



## PERFORMANCE

### General



### Database



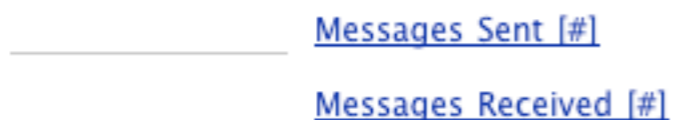
### Sessions



### View Caching



### Mailing



## ANALYTICS

### Slowest Controller Actions

AccountsController / show	417ms / 16inv
ClientsController / update	369ms / 3inv
MetricGraphsController / index	315ms / 9inv
EventsController / close_all	290ms / 1inv
MetricGraphsController / create	268ms / 29inv
EventsController / show	242ms / 14inv
ApplicationsController / action	214ms / 2inv
UsersController / edit	212ms / 3inv
SessionsController / destroy	210ms / 1inv
NotesController / create	192ms / 7inv

### Most Frequent Controller Actions

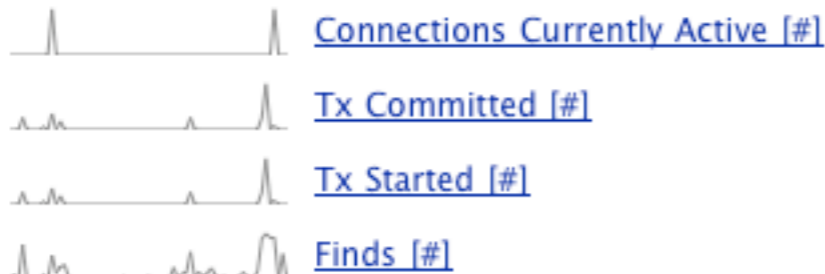
SessionsController / new	6ms / 1027inv
SourcesController / index	122ms / 46inv
SourcesController / show	126ms / 45inv
MetricGraphsController / create	268ms / 29inv
ClientsController / edit	137ms / 28inv
ApplicationsController / overview	153ms / 16inv
AccountsController / show	417ms / 16inv
EventsController / show	242ms / 14inv
SessionsController / create	110ms / 13inv
ApplicationController / index	2ms / 11inv

## PERFORMANCE

### General



### Database



### Sessions



### View Caching



### Mailing



## ANALYTICS

### Slowest Controller Actions

AccountsController / show	417ms / 16inv
ClientsController / update	369ms / 3inv
MetricGraphsController / index	315ms / 9inv
EventsController / close_all	290ms / 1inv
MetricGraphsController / create	268ms / 29inv
EventsController / show	242ms / 14inv
ApplicationsController / action	214ms / 2inv
UsersController / edit	212ms / 3inv
SessionsController / destroy	210ms / 1inv
NotesController / create	192ms / 7inv

### Most Frequent Controller Actions

SessionsController / new	6ms / 1027inv
SourcesController / index	122ms / 46inv
SourcesController / show	126ms / 45inv
MetricGraphsController / create	268ms / 29inv
ClientsController / edit	137ms / 28inv
ApplicationsController / overview	153ms / 16inv
AccountsController / show	417ms / 16inv
EventsController / show	242ms / 14inv
SessionsController / create	110ms / 13inv
ApplicationController / index	2ms / 11inv

## PERFORMANCE

### General



### Database



### Sessions



### View Caching



### Mailing



## ANALYTICS

### Slowest Controller Actions

<a href="#">AccountsController / show</a>	417ms / 16inv
<a href="#">ClientsController / update</a>	369ms / 3inv
<a href="#">MetricGraphsController / index</a>	315ms / 9inv
<a href="#">EventsController / close_all</a>	290ms / 1inv
<a href="#">MetricGraphsController / create</a>	268ms / 29inv
<a href="#">EventsController / show</a>	242ms / 14inv
<a href="#">ApplicationsController / action</a>	214ms / 2inv
<a href="#">UsersController / edit</a>	212ms / 3inv
<a href="#">SessionsController / destroy</a>	210ms / 1inv
<a href="#">NotesController / create</a>	192ms / 7inv

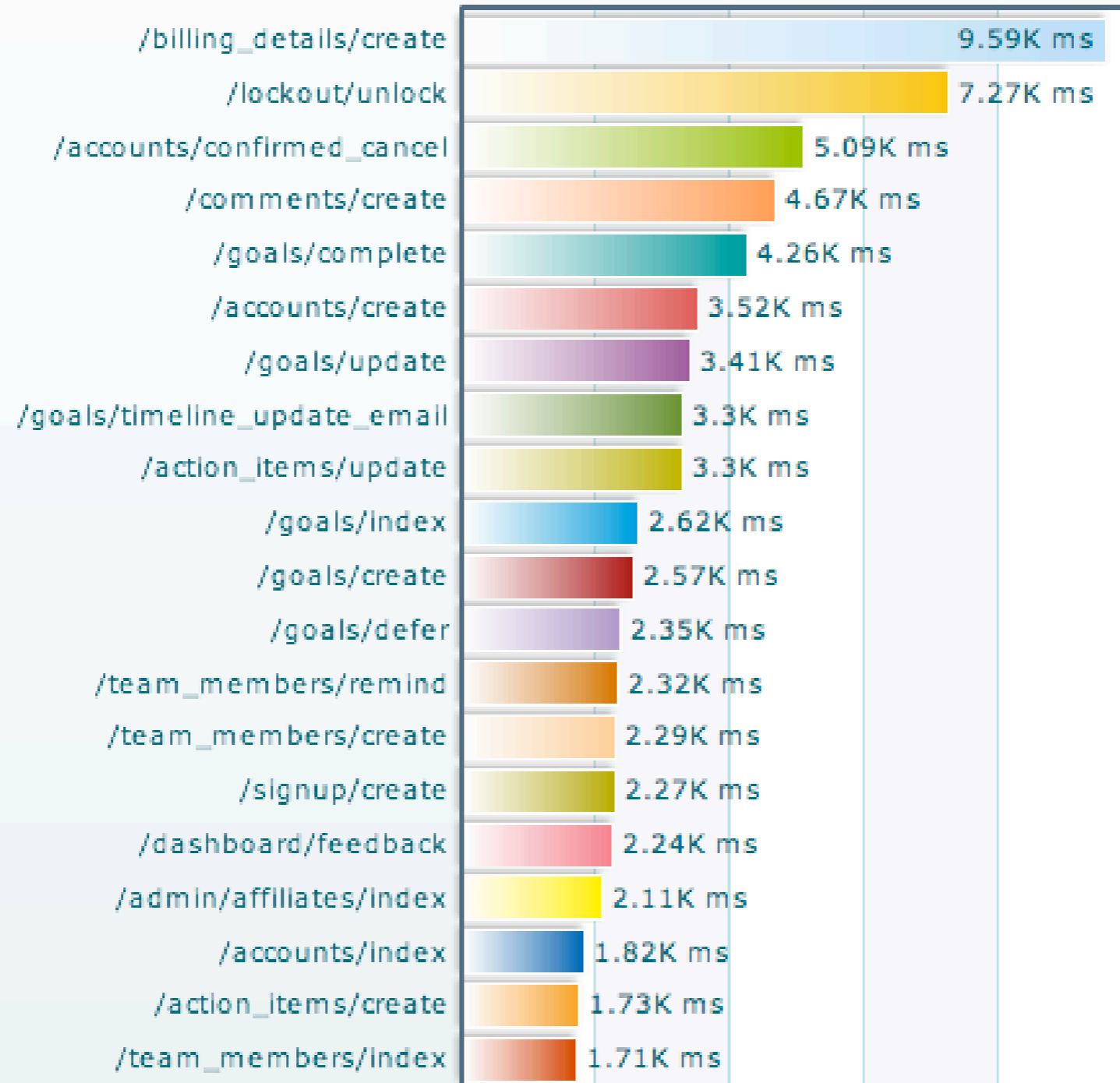
### Most Frequent Controller Actions

<a href="#">SessionsController / new</a>	6ms / 1027inv
<a href="#">SourcesController / index</a>	122ms / 46inv
<a href="#">SourcesController / show</a>	126ms / 45inv
<a href="#">MetricGraphsController / create</a>	268ms / 29inv
<a href="#">ClientsController / edit</a>	137ms / 28inv
<a href="#">ApplicationsController / overview</a>	153ms / 16inv
<a href="#">AccountsController / show</a>	417ms / 16inv
<a href="#">EventsController / show</a>	242ms / 14inv
<a href="#">SessionsController / create</a>	110ms / 13inv
<a href="#">ApplicationController / index</a>	2ms / 11inv



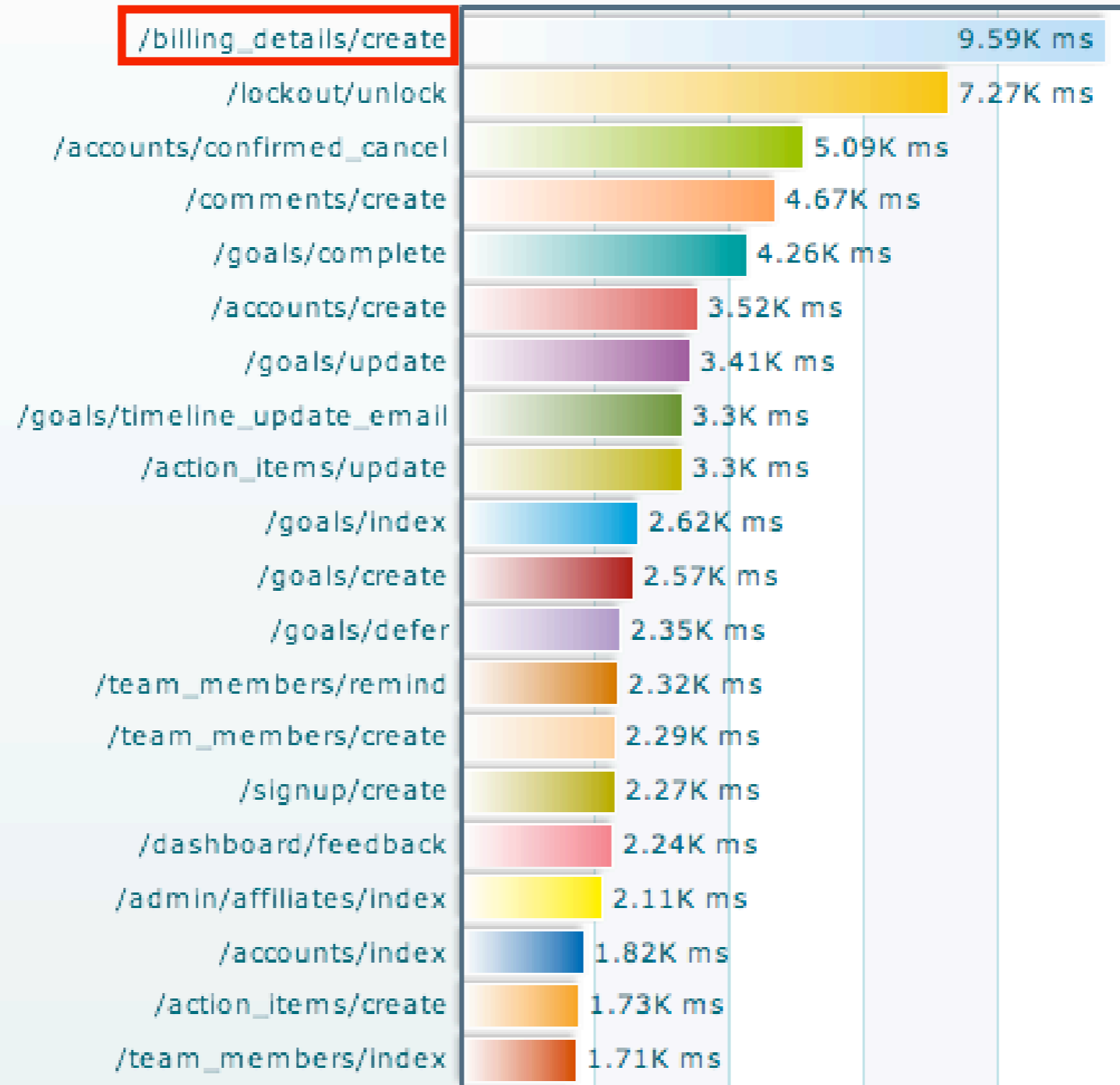
## Slowest Controller Actions

Showing top 20



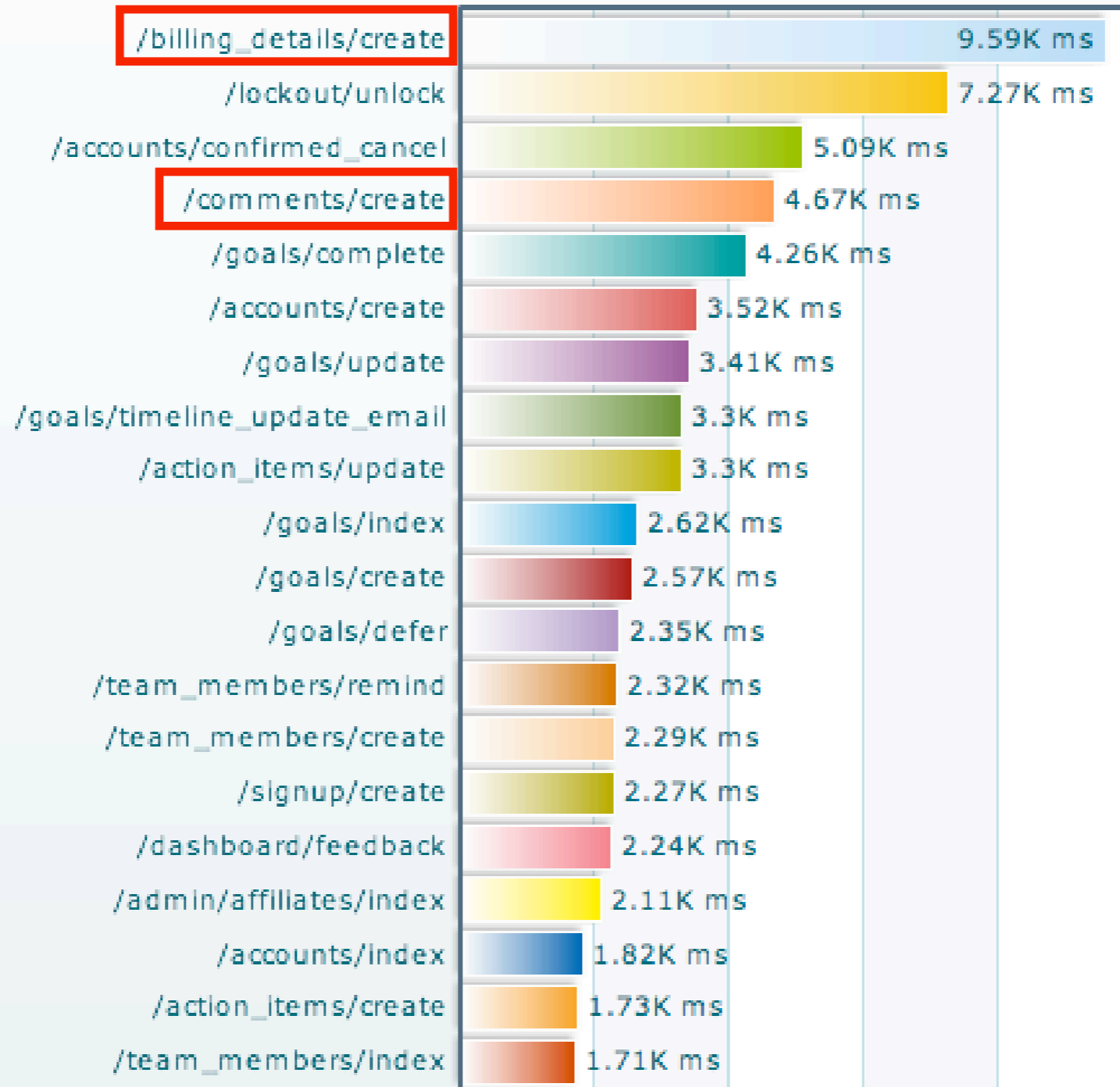
## Slowest Controller Actions

Showing top 20



## Slowest Controller Actions

Showing top 20



Grep your Logs

# Grep your Logs

Completed in 220ms (View: 72, DB: 5)

# Grep your Logs

Completed in 220ms (View: 72, DB: 5)

X-Runtime: 0.00783

# PL Analyze

```
Request Times Summary:          Count    Avg    Std Dev  Min    Max
ALL REQUESTS:                  11      0.576  0.508    0.000  1.470

ThingsController#view:         3       0.716  0.387    0.396  1.260
TeamsController#progress:      2       0.841  0.629    0.212  1.470
RssController#uber:           2       0.035  0.000    0.035  0.035
PeopleController#progress:     2       0.489  0.489    0.000  0.977
PeopleController#view:         2       0.731  0.371    0.360  1.102

Average Request Time: 0.634
Request Time Std Dev: 0.498

Slowest Request Times:
TeamsController#progress took 1.470s
ThingsController#view took 1.260s
PeopleController#view took 1.102s
PeopleController#progress took 0.977s
ThingsController#view took 0.492s
ThingsController#view took 0.396s
PeopleController#view took 0.360s
TeamsController#progress took 0.212s
RssController#uber took 0.035s
RssController#uber took 0.035s
```

**1 b. Cheat**



Caching!

# HTTP Caching

ETags

# ETags

ETag: b4c2fedde6926a5ee6ed8cd5cc995592

# ETags

ETag: b4c2fedde6926a5ee6ed8cd5cc995592

If-None-Match: b4c2fedde6926a5ee6ed8cd5cc995592

# ETags

ETag: b4c2fedde6926a5ee6ed8cd5cc995592

If-None-Match: b4c2fedde6926a5ee6ed8cd5cc995592

```
def handle_etag
  etag! [@user.id, @user.updated_at]
end
```

Last-Modified

# Last-Modified

Last-Modified: Sun, 28 Sep 2008 19:38:05 GMT



# Last-Modified

Last-Modified: Sun, 28 Sep 2008 19:38:05 GMT

If-Not-Modified-Since: Sun, 28 Sep 2008 19:38:05 GMT

# Last-Modified

Last-Modified: Sun, 28 Sep 2008 19:38:05 GMT

If-Not-Modified-Since: Sun, 28 Sep 2008 19:38:05 GMT

```
def handle_last_modified
  last_modified! @user.updated_at
end
```

# Expires

```
def add_expires  
  response.headers["Expires"] = @user.updated_at + 1.hour  
end
```

# Rails Caching

# Fragment Caching

```
<% cache [@user, "dashboard"] do %>  
  <%= render :partial=>"complex_expensive_dashboard" %>  
<% end %>
```

# Action Caching

```
cache_action :index
```

# Model Caching

```
class Comment < ActiveRecord::Base  
  acts_as_cached :ttl=>1.hour  
end
```

# Memcache

- Seriously Fast
- Seriously Scalable
- LRU Cache Expunging



# Cache Generations

There are only two hard things in  
Computer Science: cache invalidation  
and naming things

# Cache Generations

There are only two hard things in  
Computer Science: cache invalidation  
and naming things

So let's not explicitly invalidate anything

# Cache Generations

```
cache ["user", @user.id]
```

```
cache ["user_profile", @user.id]
```

# Cache Generations

```
cache ["user", @user.id, @user.generation]
```

```
cache ["user_profile", @user.id, @user.generation]
```

# Cache Generations

```
class User < ActiveRecord::Base
  before_save :increment_generation

  def increment_generation
    self.generation += 1
  end
end
```

# Cache Generations

```
class Friendship < ActiveRecord::Base
  belongs_to :owner, :class_name=>"User"
  belongs_to :friend, :class_name=>"User"
  before_save :increment_both_users

  def increment_both_users
    [owner, friend].map &:increment_generation
  end
end
```

**2. Figure out why it's  
slow**

**Never Guess**



You're **never** right

Slow Trac

# Slow Trac

- Use a new PostgreSQL driver

# Slow Trac

- Use a new PostgreSQL driver
- Use `mod_python`, not `tracd`

# Slow Trac

- Use a new PostgreSQL driver
- Use `mod_python`, not `tracd`
- It's spammers, add a better spam plugin

# Slow Trac

@project.head\_revision

# Slow Trac

```
SELECT max(rev) FROM rev;
```

# Slow Trac

```
SELECT rev FROM rev  
ORDER BY -LENGTH(rev), rev  
LIMIT 1;
```



# Slow Trac

```
SELECT max(rev::integer) FROM rev;
```

# Slow Trac

```
CREATE INDEX rev_as_INT  
ON rev(rev::integer);
```

# Performance Tests

New in 2.2

# Performance Tests

```
require 'performance/test_helper'
```

```
class SessionActionsTest < ActionController::PerformanceTest
```

# Performance Test

```
def test_fetching_index  
  get "/sessions"  
end
```

```
def test_fetching_index_all  
  get "/sessions?all=true"  
end
```

# Performance Test

```
rake test:benchmark
```

# Performance Test

`rake test:benchmark`

```
SessionActionsTest#test_fetching_index (54 ms warmup)
  process_time: 70 ms
    memory: 0.00 KB
    objects: 0
    gc_runs: 0
    gc_time: 0 ms
```

# Performance Test

```
def setup
  u = users(:koz)
  post authenticate_url, ...
  50.times do |i|
    u.sessions.create!
  end
end
```



# Performance Tests

SessionActionsTest#test\_fetching\_index (254 ms warmup)

process\_time: 270 ms

memory: 0.00 KB

objects: 0

gc\_runs: 0

gc\_time: 0 ms

# Performance Tests

```
rake test:profile
```

# Performance Tests

[SessionActionsTest#test\\_fetching\\_index\\_process\\_time\\_graph.html](#)

# Performance Tests

%Total	%Self	Total	Self	Wait	Child	Calls	Name	Line
<b>100.00%</b>	<b>0.00%</b>	<b>56.15</b>	<b>0.00</b>	<b>0.00</b>	<b>56.15</b>	<b>0</b>	<b>ActionController::RequestProfiler#profile</b>	<b><a href="#">94</a></b>
		56.15	0.00	0.00	56.15	1/1	<a href="#">ActionController::RequestProfiler#benchmark</a>	<a href="#">94</a>
		56.15	0.00	0.00	56.15	1/1	<a href="#">ActionController::RequestProfiler#profile</a>	<a href="#">94</a>
<b>100.00%</b>	<b>0.00%</b>	<b>56.15</b>	<b>0.00</b>	<b>0.00</b>	<b>56.15</b>	<b>1</b>	<b>ActionController::RequestProfiler#benchmark</b>	<b><a href="#">100</a></b>
		0.00	0.00	0.00	0.00	1/101	<a href="#">Fixnum# /</a>	<a href="#">104</a>
		0.00	0.00	0.00	0.00	2/202	<a href="#">ActionController::Integration::Runner#method_missing</a>	<a href="#">103</a>
		0.00	0.00	0.00	0.00	1/5301	<a href="#">String#%</a>	<a href="#">104</a>
		0.00	0.00	0.00	0.00	1/1	<a href="#">Integer#to_i</a>	<a href="#">103</a>
		0.00	0.00	0.00	0.00	1/2	<a href="#">Kernel#puts</a>	<a href="#">104</a>
		0.00	0.00	0.00	0.00	1/401	<a href="#">Float#to_f</a>	<a href="#">102</a>
		0.00	0.00	0.00	0.00	1/179901	<a href="#">Hash#[]</a>	<a href="#">102</a>
		56.15	0.00	0.00	56.15	1/1	<a href="#">ActionController::RequestProfiler::Sandbox#benchmark</a>	<a href="#">102</a>

# Performance Tests

%Total	%Self	Total	Self	Wait	Child	Calls	Name	Line
<b>100.00%</b>	<b>0.00%</b>	<b>56.15</b>	<b>0.00</b>	<b>0.00</b>	<b>56.15</b>	<b>0</b>	<b>ActionController::RequestProfiler#profile</b>	<b>94</b>
		56.15	0.00	0.00	56.15	1/1	<a href="#">ActionController::RequestProfiler#benchmark</a>	94
		56.15	0.00	0.00	56.15	1/1	<a href="#">ActionController::RequestProfiler#profile</a>	94
<b>100.00%</b>	<b>0.00%</b>	<b>56.15</b>	<b>0.00</b>	<b>0.00</b>	<b>56.15</b>	<b>1</b>	<b>ActionController::RequestProfiler#benchmark</b>	<b>100</b>
		0.00	0.00	0.00	0.00	1/101	<a href="#">Fixnum# /</a>	104
		0.00	0.00	0.00	0.00	2/202	<a href="#">ActionController::Integration::Runner#method_missing</a>	103
		0.00	0.00	0.00	0.00	1/5301	<a href="#">String#%</a>	104
		0.00	0.00	0.00	0.00	1/1	<a href="#">Integer#to_i</a>	103
		0.00	0.00	0.00	0.00	1/2	<a href="#">Kernel#puts</a>	104
		0.00	0.00	0.00	0.00	1/401	<a href="#">Float#to_f</a>	102
		0.00	0.00	0.00	0.00	1/179901	<a href="#">Hash#[]</a>	102
		56.15	0.00	0.00	56.15	1/1	<a href="#">ActionController::RequestProfiler::Sandbox#benchmark</a>	102



# Performance Tests

%Total	%Self	Total	Self	Wait	Child	Calls	Name	Line
<b>100.00%</b>	<b>0.00%</b>	<b>56.15</b>	<b>0.00</b>	<b>0.00</b>	<b>56.15</b>	<b>0</b>	<b>ActionController::RequestProfiler#profile</b>	<b>94</b>
		56.15	0.00	0.00	56.15	1/1	<a href="#"> ActionController::RequestProfiler#benchmark</a>	94
		56.15	0.00	0.00	56.15	1/1	<a href="#"> ActionController::RequestProfiler#profile</a>	94
<b>100.00%</b>	<b>0.00%</b>	<b>56.15</b>	<b>0.00</b>	<b>0.00</b>	<b>56.15</b>	<b>1</b>	<b>ActionController::RequestProfiler#benchmark</b>	<b>100</b>
		0.00	0.00	0.00	0.00	1/101	<a href="#"> Fixnum# /</a>	104
		0.00	0.00	0.00	0.00	2/202	<a href="#"> ActionController::Integration::Runner#method_missing</a>	103
		0.00	0.00	0.00	0.00	1/5301	<a href="#"> String#%</a>	104
		0.00	0.00	0.00	0.00	1/1	<a href="#"> Integer#to_i</a>	103
		0.00	0.00	0.00	0.00	1/2	<a href="#"> Kernel#puts</a>	104
		0.00	0.00	0.00	0.00	1/401	<a href="#"> Float#to_f</a>	102
		0.00	0.00	0.00	0.00	1/179901	<a href="#"> Hash#[]</a>	102
		56.15	0.00	0.00	56.15	1/1	<a href="#"> ActionController::RequestProfiler::Sandbox#benchmark</a>	102



# Performance Tests

SessionActionsTest#test\_fetching\_index\_all\_objects\_graph.html

# Performance Tests

%Total	%Self	Total	Self	Wait	Child	Calls	Name	Line
100.00%	0.00%	56902.00	0.00	0.00	56902.00	0	ActiveSupport::Testing::Performance::Profiler#run	17
		56902.00	1.00	0.00	56901.00	1/1	<a href="#">Kernel#_send</a>	57
		56902.00	1.00	0.00	56901.00	1/1	<a href="#">ActiveSupport::Testing::Performance::Profiler#run</a>	57
100.00%	0.00%	56902.00	1.00	0.00	56901.00	1	Kernel#_send__	0
		56901.00	1.00	0.00	56900.00	1/1	<a href="#">SessionActionsTest#test_fetching_index_all</a>	57
		56901.00	1.00	0.00	56900.00	1/1	<a href="#">Kernel#_send</a>	57
100.00%	0.00%	56901.00	1.00	0.00	56900.00	1	SessionActionsTest#test_fetching_index_all	16
		56900.00	10.00	0.00	56890.00	1/1	<a href="#">ActionController::Integration::Runner#get</a>	17



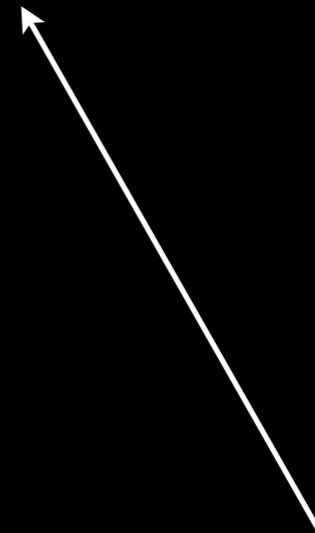
# Performance Tests

%Total	%Self	Total	Self	Wait	Child	Calls	Name	Line
100.00%	0.00%	56902.00	0.00	0.00	56902.00	0	ActiveSupport::Testing::Performance::Profiler#run	17
		56902.00	1.00	0.00	56901.00	1/1	<a href="#">Kernel#_send</a>	57
		56902.00	1.00	0.00	56901.00	1/1	<a href="#">ActiveSupport::Testing::Performance::Profiler#run</a>	57
100.00%	0.00%	56902.00	1.00	0.00	56901.00	1	Kernel#_send__	0
		56901.00	1.00	0.00	56900.00	1/1	<a href="#">SessionActionsTest#test_fetching_index_all</a>	57
		56901.00	1.00	0.00	56900.00	1/1	<a href="#">Kernel#_send</a>	57
100.00%	0.00%	56901.00	1.00	0.00	56900.00	1	SessionActionsTest#test_fetching_index_all	16
		56900.00	10.00	0.00	56890.00	1/1	<a href="#">ActionController::Integration::Runner#get</a>	17



# Performance Tests

%Total	%Self	Total	Self	Wait	Child	Calls	Name	Line
100.00%	0.00%	56902.00	0.00	0.00	56902.00	0	ActiveSupport::Testing::Performance::Profiler#run	17
		56902.00	1.00	0.00	56901.00	1/1	<a href="#">Kernel#_send</a>	57
		56902.00	1.00	0.00	56901.00	1/1	<a href="#">ActiveSupport::Testing::Performance::Profiler#run</a>	57
100.00%	0.00%	56902.00	1.00	0.00	56901.00	1	Kernel#_send__	0
		56901.00	1.00	0.00	56900.00	1/1	<a href="#">SessionActionsTest#test_fetching_index_all</a>	57
		56901.00	1.00	0.00	56900.00	1/1	<a href="#">Kernel#_send</a>	57
100.00%	0.00%	56901.00	1.00	0.00	56900.00	1	SessionActionsTest#test_fetching_index_all	16
		56900.00	10.00	0.00	56890.00	1/1	<a href="#">ActionController::Integration::Runner#get</a>	17



# Performance Tests

%Total	%Self	Total	Self	Wait	Child	Calls	Name	Line
100.00%	0.00%	56902.00	0.00	0.00	56902.00	0	ActiveSupport::Testing::Performance::Profiler#run	17
		56902.00	1.00	0.00	56901.00	1/1	<a href="#">Kernel#_send</a>	57
		56902.00	1.00	0.00	56901.00	1/1	<a href="#">ActiveSupport::Testing::Performance::Profiler#run</a>	57
100.00%	0.00%	56902.00	1.00	0.00	56901.00	1	Kernel#_send__	0
		56901.00	1.00	0.00	56900.00	1/1	<a href="#">SessionActionsTest#test_fetching_index_all</a>	57
		56901.00	1.00	0.00	56900.00	1/1	<a href="#">Kernel#_send</a>	57
100.00%	0.00%	56901.00	1.00	0.00	56900.00	1	SessionActionsTest#test_fetching_index_all	16
		56900.00	10.00	0.00	56890.00	1/1	<a href="#">ActionController::Integration::Runner#get</a>	17



# 3. Make it Faster

```
<%= image_tag "red.png" %>
```

```

```

```
<%= image_tag "onepixel.gif" %>
```







# N+1 Queries

```
<% @user.posts.each do |post| %>  
  <p><%= link_to post.title, post_url(post) %></p>  
<% end %>
```

# N+1 Queries

```
User.find(params[:id], :include=>[:posts])
```

# N+1 Queries

```
<% @user.posts.each do |post| %>  
  <p>  
    <%= link_to post.title, post_url(post) %>  
    by  
    <%= post.author.name %>  
  </p>  
<% end %>
```

# Garbage Collection

PageShowTest#test\_show (246 ms warmup)

process\_time: 183 ms

memory: 1696.26 KB

objects: 116280

gc\_runs: 0

gc\_time: 42 ms

22% in GC

# Garbage Collection

<http://github.com/skaes/railsbench>

# Garbage Collection

```
# greatly increase initial heap slots available (60x more)
RUBY_HEAP_MIN_SLOTS=600000
# ensure lots of heap slots are freed after GC (25x more)
RUBY_HEAP_FREE_MIN=100000
# and don't run GC unless we're over 60MB heap (7.5x more)
RUBY_GC_MALLOC_LIMIT=60000000
```

# Conclusion

**Fix Frontend  
Performance First**



**Use HTTP**

**Cache with memcache**

**Profile, don't guess**

**Don't forget GC when  
profiling**

# Questions?

Michael Koziarski  
[michael@koziarski.com](mailto:michael@koziarski.com)

