



Web Framework Performance

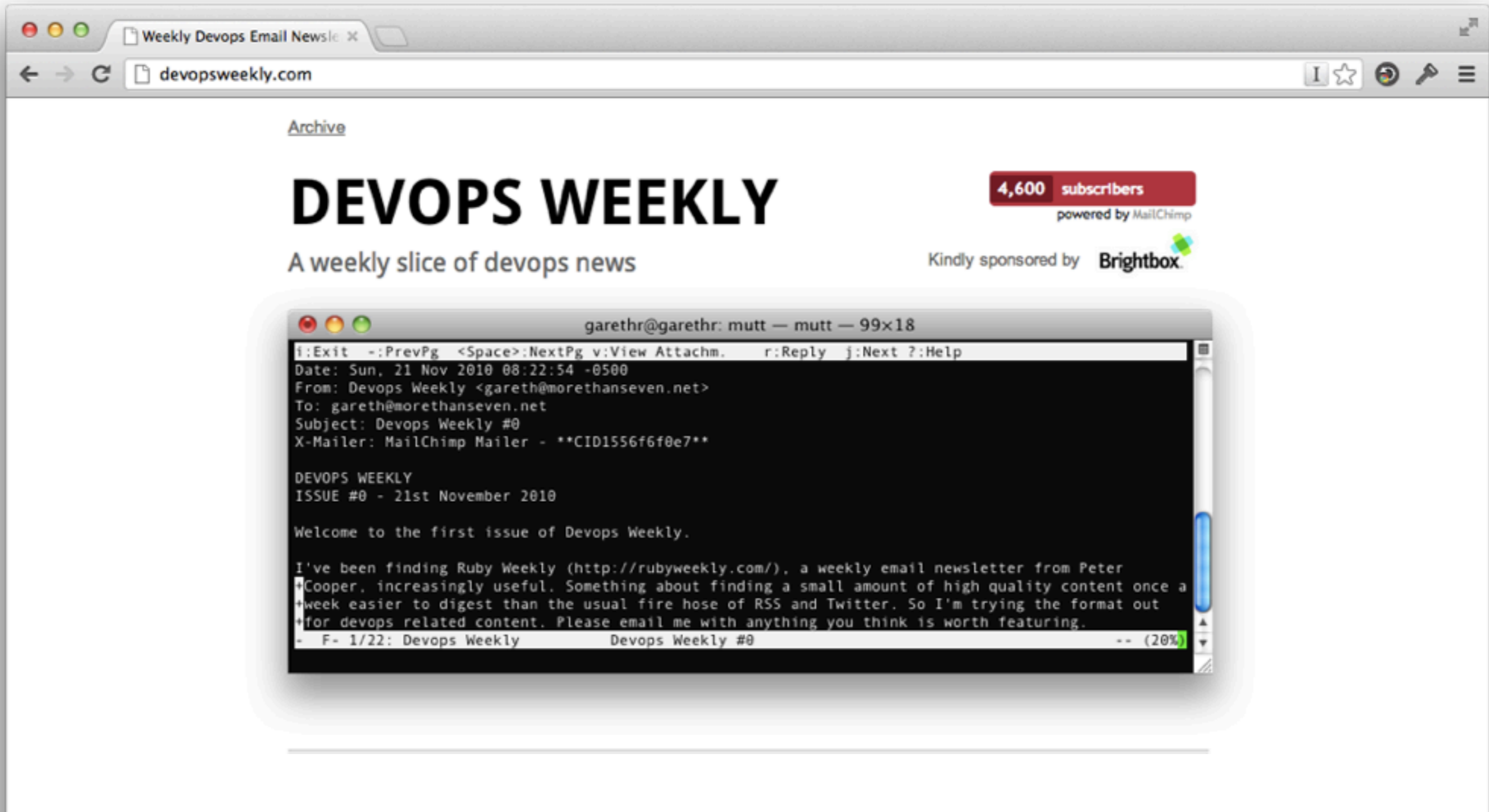
Examples from Django and Rails

QConSF 9th November 2012

Me



Gareth Rushgrove



Curate devopsweekly.com

Morethanseven

The blog of Gareth Rushgrove



Meet Mandrill - Mandrill is a new transactional email service from MailChimp. Sign up today.

ads via Carbon

Shipping

OCT 21ST, 2012 | [0 COMMENTS AND 0 REACTIONS](#)

Last week we shipped [GOV.UK](#). Over the last year we've built a team to build a website. Now we're busy building a culture too. I've got so much that needs writing up about everything we've been up to. Hopefully I'll make a start in the next week or so.

Tale of a Grok Pattern

Blog at morethanseven.net



Work at UK Government Digital Service

CIVIL SERVICE — CLUB —

MEMBERSHIP AVAILABLE TO ALL
WORKING & RETIRED CIVIL SERVANTS
— SEE INCL

I am a Civil Servant

Regular Releases Reduce Risk

digital.cabinetoffice.gov.uk/2012/11/02/regular-releases-reduce-risk/

Government Digital Service

Home About GDS GDS projects Working at GDS Contact us Cookies GOV UK Cabinet Office homepage

Regular Releases Reduce Risk

by Gareth Rushgrove on 02/11/2012

GOV.UK came out of beta just over two weeks ago. In that time we've released over 100 updates. I want to talk a little about how we do that, but mainly to focus on why this approach makes for more stability and less risk for the service.

Before we begin I just want to say thank you to one of our Cabinet Office colleagues who has been instrumental in getting me to write this blog post.

@GDSTeam on Twitter

The UK's digital landscape. Blog post by Nick Breeze
digital.cabinetoffice.gov.uk/2012/11/06/the-uk-digital-landscape/
@GDSTeam #govdigstrat 4 hours ago

The Digital Efficiency report. Blog post by Andrew Greenway

Developer, Operations, Product Guy

Introduction

(what's the problem)



Slow(er) languages...



Slow(er) languages...



...don't mean slow applications



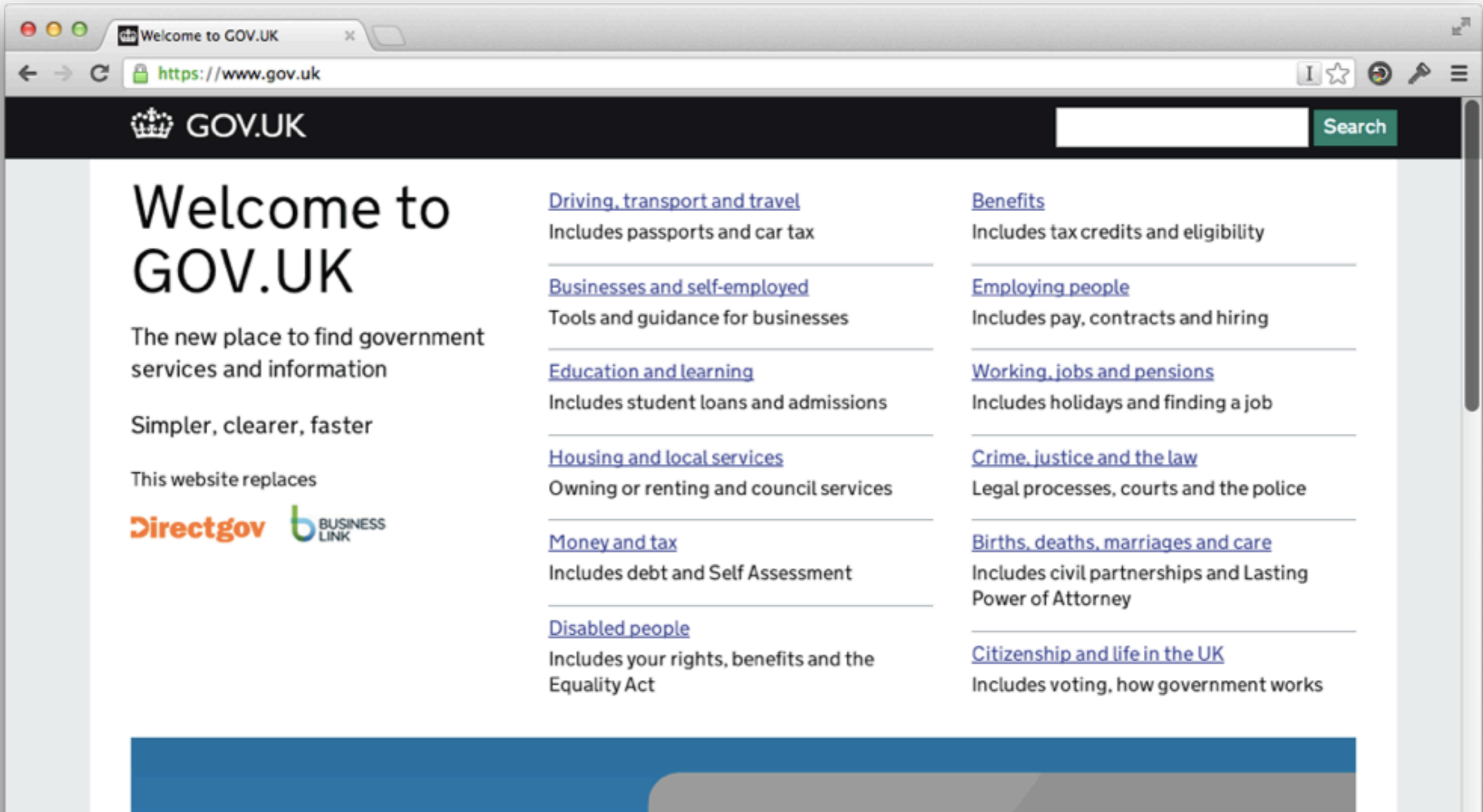
...don't *have to* mean slow applications



Frameworks can help

django

Frameworks can help



A real example

Release-o-tron Rails

localhost:3000/releases

Home Releases Applications

Application Releases

what and when [New release](#)

Id	App	Released
51848956	frontend	3 Nov 2012 17:11PM
113629430	admin	3 Nov 2012 17:11PM
281110143	frontend	3 Nov 2012 17:11PM
298486374	api	3 Nov 2012 17:11PM
374648174	api	3 Nov 2012 17:11PM
507103779	admin	3 Nov 2012 17:11PM
638205575	frontend	3 Nov 2012 17:11PM
980190962	frontend	3 Nov 2012 17:11PM
991788158	api	3 Nov 2012 17:11PM
1018350795	frontend	3 Nov 2012 17:11PM

A sample application

49ms

Before

6ms

After

- Analyze your application
- Know your framework
- Cache everywhere
- Instrument everything
- Don't just think about development

This presentation

Analyze
(count everything)



1. From the browser

The screenshot shows a web browser window with the URL `localhost:3000/releases`. The page content includes a navigation bar with 'Home', 'Releases', and 'Applications' links. The main heading is 'Application Releases what and when' with a 'New release' button. Below the browser, the YSlow performance tool interface is visible, showing a 'Grade A' overall performance score of 92. The tool lists various performance metrics and their scores:

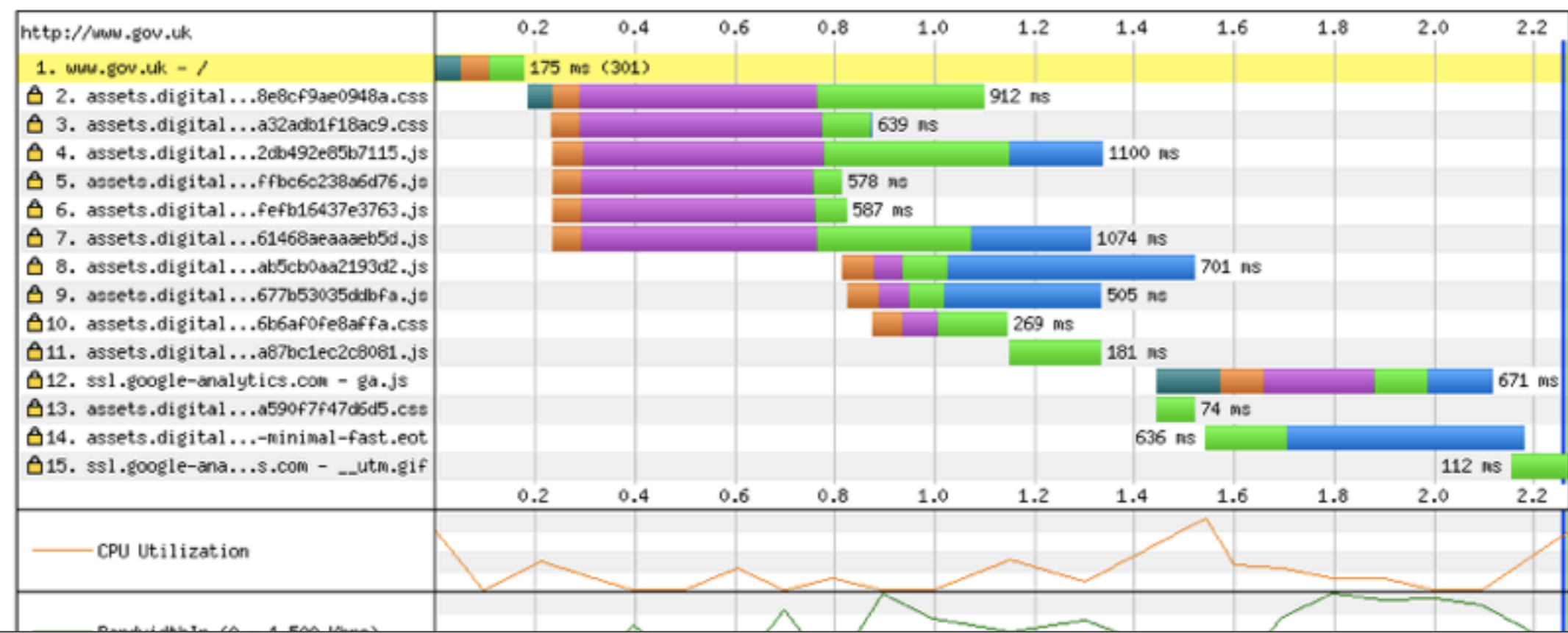
- Make fewer HTTP requests: **A**
- Use a Content Delivery Network (CDN): **C**
- Avoid empty src or href: **A**
- Add Expires headers: **E**
- Compress components with gzip: **D**
- Put CSS at top: **A**
- Put JavaScript at bottom: **A**
- Avoid CSS expressions: **A**

The 'Make fewer HTTP requests' section is expanded, showing a 'Grade A' score and a detailed explanation: 'Decreasing the number of components on a page reduces the number of HTTP requests required to render the page, resulting in faster page loads. Some ways to reduce the number of components include: combine files, combine multiple scripts into one script, combine multiple CSS files into one style sheet, and use CSS Sprites and image maps.' A '»Read More' link is provided for further information.

YSlow

Load Time	First Byte	Start Render	DOM Elements	Result (error code)	Document Complete			Fully Loaded		
					Time	Requests	Bytes In	Time	Requests	Bytes In
2.255s	0.176s	2.265s	206	0	2.255s	15	206 KB	2.269s	15	206 KB

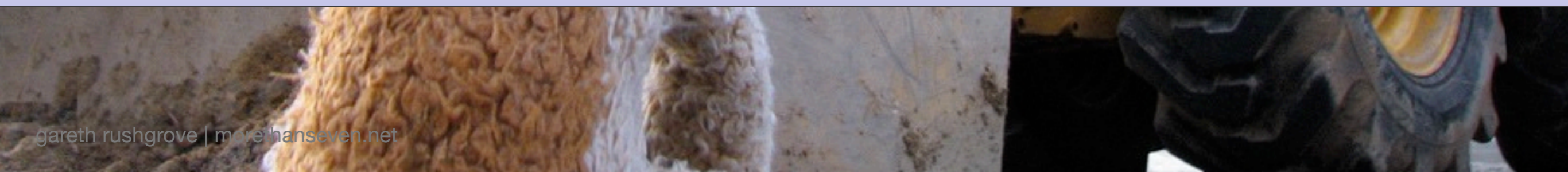
Waterfall View



webpagetest.org



2. From the code



Release-o-tron Django

localhost:8000

Admin Releases

Application Releases what and when

Id	App	Released
10	Admin	Oct. 29, 2012
9	Frontend	Oct. 29, 2012
8	Frontend	Oct. 29, 2012
7	Admin	Oct. 29, 2012
6	API	Oct. 29, 2012
5	Frontend	Oct. 29, 2012
4	API	Oct. 29, 2012
3	API	Oct. 29, 2012
2	Frontend	Oct. 29, 2012
1	Frontend	Oct. 29, 2012

POWERED BY **django**

Django debug toolbar



Release-o-tron Django

localhost:8000

Admin Releases

Application Releases what and when

Id	App	Released
10	Admin	Oct. 29, 2012
9	Frontend	Oct. 29, 2012
8	Frontend	Oct. 29, 2012
7	Admin	Oct. 29, 2012
6	API	Oct. 29, 2012
5	Frontend	Oct. 29, 2012
4	API	Oct. 29, 2012
3	API	Oct. 29, 2012
2	Frontend	Oct. 29, 2012
1	Frontend	Oct. 29, 2012

POWERED BY **django**

Hide »

- Versions
DJANGO 1.4.2
- Time
CPU: 190.00ms (225.29ms)
- Settings
- HTTP Headers
- Request Vars
- Templates
- SQL
1 QUERY IN 5.17ms
- Signals
13 RECEIVERS FROM 12 SIGNALS
- Logging
0 MESSAGES
- Memcache
11 CALLS IN 4.89ms
- Statsd

Django debug toolbar



The screenshot shows a web browser window with the Django Debug Toolbar (DDT) active. The main panel displays 'Resource Usage' with the following data:

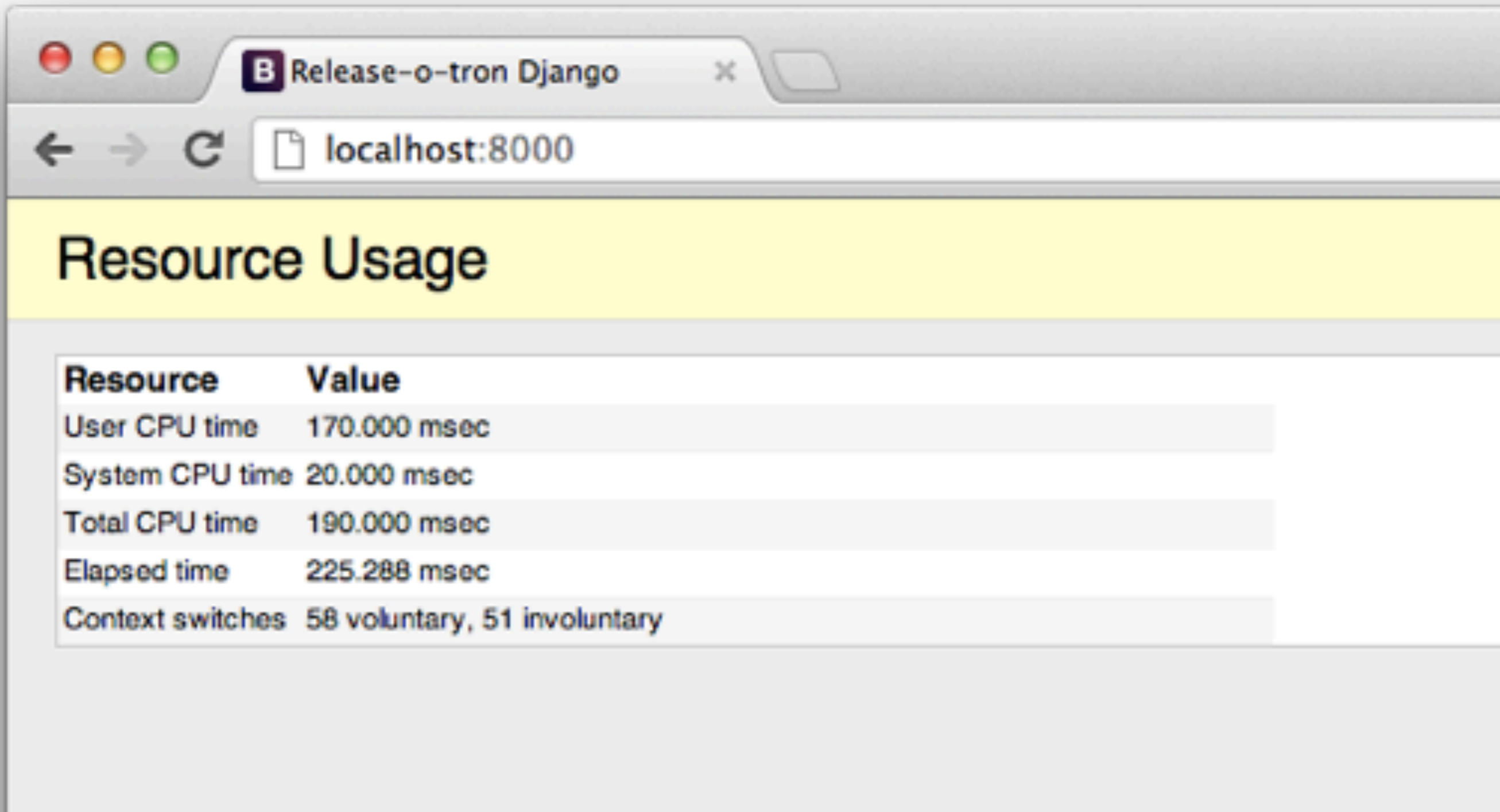
Resource	Value
User CPU time	170.000 msec
System CPU time	20.000 msec
Total CPU time	190.000 msec
Elapsed time	225.288 msec
Context switches	58 voluntary, 51 involuntary

The sidebar on the right contains the following panels:

- Hide »
- Versions: DJANGO 1.4.2
- Time: CPU: 190.00ms (225.29ms)
- Settings
- HTTP Headers
- Request Vars
- Templates
- SQL: 1 QUERY IN 5.17MS
- Signals: 13 RECEIVERS FROM 12 SIGNALS
- Logging: 0 MESSAGES
- Memcache: 11 CALLS IN 4.89MS
- Statsd

Django debug toolbar





The screenshot shows a web browser window with a single tab titled "Release-o-tron Django". The address bar contains "localhost:8000". Below the address bar, the page content is displayed on a yellow background with the heading "Resource Usage". Underneath, a table lists various resource metrics and their values.

Resource	Value
User CPU time	170.000 msec
System CPU time	20.000 msec
Total CPU time	190.000 msec
Elapsed time	225.288 msec
Context switches	58 voluntary, 51 involuntary

Django debug toolbar



```

localhost:8000/?prof
localhost:8000/?prof

77 function calls (68 primitive calls) in 0.001 CPU seconds

Ordered by: internal time, call count

ncalls  tottime  percall  cumtime  percall  filename:lineno(function)
1      0.000    0.000    0.001    0.001  /home/vagrant/env/django/lib/python2.6/site-packages/django/views/generic/base.py:44(view)
1      0.000    0.000    0.000    0.000  /home/vagrant/env/django/lib/python2.6/site-packages/django/views/generic/base.py:19(__init__)
1      0.000    0.000    0.000    0.000  /home/vagrant/env/django/lib/python2.6/site-packages/django/views/generic/base.py:58(dispatch)
1      0.000    0.000    0.000    0.000  /home/vagrant/env/django/lib/python2.6/site-packages/django/views/generic/list.py:17(get_queryset)
1      0.000    0.000    0.000    0.000  /home/vagrant/env/django/lib/python2.6/site-packages/django/db/models/sql/query.py:235(clone)
12/4   0.000    0.000    0.000    0.000  /usr/lib/python2.6/copy.py:144(deepcopy)
1      0.000    0.000    0.000    0.000  /home/vagrant/env/django/lib/python2.6/site-packages/django/http/__init__.py:569(__init__)
12     0.000    0.000    0.000    0.000  /usr/lib/python2.6/copy.py:261(_keep_alive)
3      0.000    0.000    0.000    0.000  /home/vagrant/env/django/lib/python2.6/site-packages/django/utils/datastructures.py:108(__init__)
1      0.000    0.000    0.000    0.000  /home/vagrant/env/django/lib/python2.6/site-packages/django/views/generic/list.py:113(get)
2/1   0.000    0.000    0.000    0.000  /usr/lib/python2.6/copy.py:232(_deepcopy_tuple)
2      0.000    0.000    0.000    0.000  /home/vagrant/env/django/lib/python2.6/site-packages/django/utils/tree.py:55(__deepcopy__)
1      0.000    0.000    0.000    0.000  /home/vagrant/env/django/lib/python2.6/site-packages/django/db/models/query.py:870(_clone)
1      0.000    0.000    0.000    0.000  /home/vagrant/env/django/lib/python2.6/site-packages/django/views/generic/list.py:73(get_context_object_name)
3      0.000    0.000    0.000    0.000  /home/vagrant/env/django/lib/python2.6/site-packages/django/utils/datastructures.py:103(__new__)
2      0.000    0.000    0.000    0.000  /home/vagrant/env/django/lib/python2.6/site-packages/django/utils/datastructures.py:200(copy)
1      0.000    0.000    0.000    0.000  /home/vagrant/env/django/lib/python2.6/site-packages/django/views/generic/list.py:126(get_template_names)
1      0.000    0.000    0.000    0.000  /home/vagrant/env/django/lib/python2.6/site-packages/django/utils/datastructures.py:127(__deepcopy__)
5      0.000    0.000    0.000    0.000  /usr/lib/python2.6/copy.py:224(_deepcopy_list)
1      0.000    0.000    0.000    0.000  /home/vagrant/env/django/lib/python2.6/site-packages/django/http/__init__.py:609(__setitem__)
1      0.000    0.000    0.000    0.000  /home/vagrant/env/django/lib/python2.6/site-packages/django/views/generic/base.py:89(render_to_response)
1      0.000    0.000    0.000    0.000  /home/vagrant/env/django/lib/python2.6/site-packages/django/template/response.py:140(__init__)
1      0.000    0.000    0.000    0.000  /usr/lib/python2.6/copy.py:300(_reconstruct)
3      0.000    0.000    0.000    0.000  /home/vagrant/env/django/lib/python2.6/site-packages/django/http/__init__.py:594(_convert_to_ascii)
1      0.000    0.000    0.000    0.000  /home/vagrant/env/django/lib/python2.6/site-packages/django/views/generic/list.py:84(get_context_data)
1      0.000    0.000    0.000    0.000  /home/vagrant/env/django/lib/python2.6/site-packages/django/template/response.py:12(__init__)
1      0.000    0.000    0.000    0.000  /home/vagrant/env/django/lib/python2.6/site-packages/django/db/models/query.py:32(__init__)
1      0.000    0.000    0.000    0.000  /home/vagrant/env/django/lib/python2.6/site-packages/django/http/__init__.py:700(_set_content)
1      0.000    0.000    0.000    0.000  /home/vagrant/env/django/lib/python2.6/site-packages/django/utils/encoding.py:104(smart_str)
2      0.000    0.000    0.000    0.000  /home/vagrant/env/django/lib/python2.6/site-packages/django/utils/functional.py:182(inner)
2      0.000    0.000    0.000    0.000  /home/vagrant/env/django/lib/python2.6/site-packages/django/utils/tree.py:18(__init__)
1      0.000    0.000    0.000    0.000  /home/vagrant/env/django/lib/python2.6/site-packages/django/template/response.py:127(_set_content)
2      0.000    0.000    0.000    0.000  /home/vagrant/env/django/lib/python2.6/site-packages/django/utils/datastructures.py:164(keys)
1      0.000    0.000    0.000    0.000  /home/vagrant/env/django/lib/python2.6/site-packages/django/views/generic/base.py:100(get_template_names)

```

Profiling middleware



Release-o-tron Rails

localhost:3000/releases

Home Releases Applications

Application Releases

Id	App
51848956	frontend
113629430	admin
281110143	frontend
298486374	api
374648174	api
507103779	admin
638205575	frontend
980190962	frontend
991788158	api
1018350795	frontend

/releases localhost on Tue, 06 Nov 2012 19:43:09 GMT

	duration (ms)	with children (ms)	from start (ms)	query time (ms)
GET http://localhost:3000/releases	27.1	245.9	+0.0	1 sql 0.1
Executing action: index	187.1	218.8	+26.0	8 sql 2.8
Rendering: releases/index	29.3	29.3	+212.0	
Rendering: layouts/application	2.4	2.4	+241.0	

[hide time with children](#) 1.2 % in sql

client event	duration (ms)	from start (ms)
Fetch	0.0	+0.0
Navigation	0.0	+0.0
Domain Lookup	1.0	+2.0
Request	0.0	+3.0
Connect	0.0	+3.0
Response	2.0	+554.0
Unload Event	0.0	+556.0
domLoading		+560.0
Dom Content Loaded Event	6.0	+914.0
domInteractive		+914.0
Load Event	1.0	+945.0
domComplete		+945.0
First Paint Time		+971.0
First Paint After Load Time		+971.0

[share](#) [hide trivia](#)

3 Nov 2012 17:11PM

rack-mini-profiler



2.9 ms

0.9 ms

245.9 ms

/releases

localhost on Tue, 06 Nov 2012 19:43:09 GMT

	duration (ms)	with children (ms)	from start (ms)	query time (ms)
GET http://localhost:3000/releases	27.1	245.9	+0.0	1 sql 0.1
Executing action: index	187.1	218.8	+26.0	8 sql 2.8
Rendering: releases/index	29.3	29.3	+212.0	
Rendering: layouts/application	2.4	2.4	+241.0	

[hide time with children](#)

1.2 @ in sql

client event	duration (ms)	from start (ms)
Fetch	0.0	+0.0
Navigation	0.0	+0.0
Domain Lookup	1.0	+2.0
Request	0.0	+3.0
Connect	0.0	+3.0
Response	2.0	+554.0
Unload Event	0.0	+556.0
domLoading		+560.0
Dom Content Loaded Event	6.0	+914.0
domInteractive		+914.0
Load Event	1.0	+945.0

rack-mini-profiler details



3 Nov 2012 17:11PM

Release-o-tron Rails


localhost:3000/releases

[X] Rack::Insight (req# 60) Rails 3.2.8 109.52ms RequestVariables (1) 0 Queries (0.00ms) 0 AR Objects Cache: 1.00ms (2 calls) Templates: 40.00ms Log (5) -588 KB Δ, 75252 KB total

Application Releases

what and when [New release](#)

Id	App	Released
51848956	frontend	3 Nov 2012 17:11PM
113629430	admin	3 Nov 2012 17:11PM
281110143	frontend	3 Nov 2012 17:11PM
298486374	api	3 Nov 2012 17:11PM
374648174	api	3 Nov 2012 17:11PM
507103779	admin	3 Nov 2012 17:11PM
638205575	frontend	3 Nov 2012 17:11PM
980190962	frontend	3 Nov 2012 17:11PM
991788158	api	3 Nov 2012 17:11PM
1018350795	frontend	3 Nov 2012 17:11PM



Rack Insight



113629430	admin	3 Nov 2012 17:11PM
281110143	frontend	3 Nov 2012 17:11PM
298486374	api	3 Nov 2012 17:11PM
374648174	api	3 Nov 2012 17:11PM
507103779	admin	3 Nov 2012 17:11PM
638205575	frontend	3 Nov 2012 17:11PM
980190962	frontend	3 Nov 2012 17:11PM
991788158	api	3 Nov 2012 17:11PM
1018350795	frontend	3 Nov 2012 17:11PM



Resource Usage

60: GET /releases Close

Key	Value
User CPU time	70.00ms
System CPU time	20.00ms
Total CPU time	90.00ms
Elapsed time	109.52ms

Rack Insight



980190962	frontend	3 Nov 2012 17:11PM
991788158	api	3 Nov 2012 17:11PM
1018350795	frontend	3 Nov 2012 17:11PM



Edit: Controller | [Partials \(0\)](#) | [Stylesheets \(1\)](#) | [Javascripts \(1\)](#)

Show: [Assigns \(1\)](#) | [Session \(1\)](#) | [Cookies \(11\)](#) | [Params \(2\)](#) | [Filters](#) | [Routes](#) | [Env](#) | [Queries \(6\)](#) DB (1.341ms) | [Log \(0\)](#) | [General Debug](#)

Queries

```
UNKNOWN (trace)
SHOW FULL FIELDS FROM releases
SCHEMA (0.001ms)
UNKNOWN (trace)
SHOW CREATE TABLE releases
SCHEMA (0.000ms)
SELECT (trace)
SELECT releases.* FROM releases
Release Load (0.000ms)
UNKNOWN (trace)
SHOW CREATE TABLE apps
SCHEMA (0.000ms)
UNKNOWN (trace)
SHOW FULL FIELDS FROM apps
SCHEMA (0.000ms)
SELECT (trace)
SELECT apps.* FROM apps WHERE apps.id IN (1, 3, 2)
App Load (0.000ms)
```

Rails footnotes



New Relic Developer Mode

localhost:3000/newrelic/show_sample_detail?id=30499920

New Relic RPM™ » Home

Developer Mode Monitor your Rails Application in production. [Learn more.](#)

URL: /releases
 Controller: ReleasesController
 Action: index
 Start Time: 11:50:21
 Duration: 215 ms
 CPU Burn: ms

[Summary] [Details] [SQL]

Expand All Collapse All

Metric	Timestamp seconds	Duration ms	Exclusive ms	
ReleasesController#index	0.001	214	109	
get	0.003	33	33	
SQL - SHOW	0.081	1	1	SQL
SQL - SHOW	0.083	0	0	SQL
SQL - SHOW	0.084	0	0	SQL
▶ Release#find_by_sql	0.084	30	29	SQL
SQL - SHOW	0.151	0	0	SQL
SQL - SHOW	0.152	0	0	SQL
SQL - SHOW	0.152	1	1	SQL
▶ App#find_by_sql	0.153	4	4	SQL
set	0.158	1	1	

New Relic development mode



```
Started GET "/assets/rails.png" for 10.0.2.2 at 2012-10-26 03:55:00 -0700
Connecting to database specified by database.yml
Served asset /rails.png - 200 OK (5ms)

Started GET "/rails/info/properties" for 10.0.2.2 at 2012-10-26 03:55:07 -0700
Processing by Rails::InfoController#properties as */*
  Rendered inline template (2.9ms)
Completed 200 OK in 33ms (Views: 32.2ms | ActiveRecord: 0.0ms)

Started GET "/rails/info/properties" for 10.0.2.2 at 2012-10-26 03:55:11 -0700
Processing by Rails::InfoController#properties as */*
  Rendered inline template (0.3ms)
Completed 200 OK in 1ms (Views: 0.6ms | ActiveRecord: 0.0ms)

Started GET "/rails/info/properties" for 10.0.2.2 at 2012-10-26 03:55:13 -0700
Processing by Rails::InfoController#properties as */*
  Rendered inline template (0.3ms)
Completed 200 OK in 1ms (Views: 0.6ms | ActiveRecord: 0.0ms)

Started GET "/bob" for 10.0.2.2 at 2012-10-26 03:55:32 -0700

ActionController::RoutingError (No route matches [GET] "/bob"):
  actionpack (3.2.8) lib/action_dispatch/middleware/debug_exceptions.rb:21:in `call'
  actionpack (3.2.8) lib/action_dispatch/middleware/show_exceptions.rb:56:in `call'
  railties (3.2.8) lib/rails/rack/logger.rb:26:in `call_app'
  railties (3.2.8) lib/rails/rack/logger.rb:16:in `call'
  actionpack (3.2.8) lib/action_dispatch/middleware/request_id.rb:22:in `call'
  rack (1.4.1) lib/rack/methodoverride.rb:21:in `call'
  rack (1.4.1) lib/rack/runtime.rb:17:in `call'
  activesupport (3.2.8) lib/active_support/cache/strategy/local_cache.rb:72:in `call'
  rack (1.4.1) lib/rack/lock.rb:15:in `call'
  actionpack (3.2.8) lib/action_dispatch/middleware/static.rb:62:in `call'
  railties (3.2.8) lib/rails/engine.rb:479:in `call'
  railties (3.2.8) lib/rails/application.rb:223:in `call'
  rack (1.4.1) lib/rack/content_length.rb:14:in `call'
```

3. Logs

```
  Rendered /usr/lib/ruby/gems/1.9.1/gems/actionpack-3.2.8/lib/action_dispatch/middleware/templates/rescues/routing_error.erb within re
scues/layout (0.6ms)
log/development.log
```

Request duration - by sum	Hits	Sum	Mean	StdDev	Min	Max	95 %tile
ReleasesController#index.HTML	153	23.92s	0.16s	0.26s	0.01s	1.74s	0.01s-1.08s
SampleController#index.HTML	27	1.43s	0.05s	0.06s	0.01s	0.25s	0.01s-0.26s
ReleasesController#show.HTML	8	1.02s	0.13s	0.13s	0.02s	0.39s	0.02s-0.40s
AppsController#index.HTML	11	1.00s	0.09s	0.10s	0.02s	0.36s	0.02s-0.37s
AppsController#new.HTML	6	0.43s	0.07s	0.04s	0.02s	0.15s	0.02s-0.15s
ReleasesController#new.HTML	6	0.20s	0.03s	0.01s	0.02s	0.05s	0.02s-0.05s
ReleasesController#edit.HTML	2	0.17s	0.08s	0.03s	0.06s	0.11s	0.06s-0.11s
ReleasesController#update.HTML	1	0.14s	0.14s	0.00s	0.14s	0.14s	0.14s-0.14s
AppsController#destroy.HTML	3	0.14s	0.05s	0.01s	0.03s	0.05s	0.03s-0.05s
ReleasesController#index.*/*	1	0.07s	0.07s	0.00s	0.07s	0.07s	0.07s-0.07s
AppsController#create.HTML	1	0.06s	0.06s	0.00s	0.06s	0.06s	0.06s-0.06s
Rails::InfoController#properties.*/*	3	0.04s	0.01s	0.02s	0.00s	0.03s	0.00s-0.03s
AppsController#show.HTML	1	0.03s	0.03s	0.00s	0.03s	0.03s	0.03s-0.03s
ReleasesController#create.HTML	1	0.02s	0.02s	0.00s	0.02s	0.02s	0.02s-0.02s

Request duration - by mean	Hits	Sum	Mean	StdDev	Min	Max	95 %tile
ReleasesController#index.HTML	153	23.92s	0.16s	0.26s	0.01s	1.74s	0.01s-1.08s
ReleasesController#update.HTML	1	0.14s	0.14s	0.00s	0.14s	0.14s	0.14s-0.14s
ReleasesController#show.HTML	8	1.02s	0.13s	0.13s	0.02s	0.39s	0.02s-0.40s
AppsController#index.HTML	11	1.00s	0.09s	0.10s	0.02s	0.36s	0.02s-0.37s
ReleasesController#edit.HTML	2	0.17s	0.08s	0.03s	0.06s	0.11s	0.06s-0.11s
AppsController#new.HTML	6	0.43s	0.07s	0.04s	0.02s	0.15s	0.02s-0.15s
ReleasesController#index.*/*	1	0.07s	0.07s	0.00s	0.07s	0.07s	0.07s-0.07s
AppsController#create.HTML	1	0.06s	0.06s	0.00s	0.06s	0.06s	0.06s-0.06s
SampleController#index.HTML	27	1.43s	0.05s	0.06s	0.01s	0.25s	0.01s-0.26s
AppsController#destroy.HTML	3	0.14s	0.05s	0.01s	0.03s	0.05s	0.03s-0.05s
AppsController#show.HTML	1	0.03s	0.03s	0.00s	0.03s	0.03s	0.03s-0.03s
ReleasesController#new.HTML	6	0.20s	0.03s	0.01s	0.02s	0.05s	0.02s-0.05s
ReleasesController#create.HTML	1	0.02s	0.02s	0.00s	0.02s	0.02s	0.02s-0.02s
Rails::InfoController#properties.*/*	3	0.04s	0.01s	0.02s	0.00s	0.03s	0.00s-0.03s

View rendering time - by sum	Hits	Sum	Mean	StdDev	Min	Max	95 %tile
ReleasesController#index.HTML	146	13.02s	0.09s	0.25s	0.00s	1.59s	0.00s-1.39s

AppsController#show.HTML	1	0.01s	0.01s	0.00s	0.01s	0.01s	0.01s-0.01s
--------------------------	---	-------	-------	-------	-------	-------	-------------

Request log analyzer



	Mean		StdDev		Min		Max		95 %tile	
	0.16s		0.26s		0.01s		1.74s		0.01s-1.08s	

Request log analyzer details



2. vagrant@lucid64: ~/django_sample (ssh)

view	method	status	count	minimum	maximum	mean	stdev	queries	querytime
django.contrib.admin.options.changelist_view	GET	200	1	0.140	0.140	0.140	0	3	0
django.contrib.admin.sites.118n_javascript	GET	200	1	0.050	0.050	0.050	0	0	0
django.contrib.admin.sites.index	GET	200	7	0.060	0.190	0.140	0.040	1	0
django.contrib.admin.sites.index	POST	302	2	0.240	0.380	0.310	0.100	6	0
django.contrib.admin.sites.index	GET	500	3	0.130	0.160	0.143	0.020	0	0
django.contrib.staticfiles.views.serve	GET	200	42	0	0.040	0.005	0.010	0	0
django.contrib.s	GET	404	4	0.050	0.060	0.055	0.010	0	0
django.v	GET	304	20	0	0.030	0.004	0.010	0	0

Django timelog



Know your framework

(use the parts you need)



1. Disable what you don't need

```
require "rails/all"
```

```
in config/application.rb
```

Don't import rails



```
require "action_controller/railtie"  
require "rails/test_unit/railtie"  
require "sprockets/railtie"
```

in config/application.rb

Just the bits you need



```
MIDDLEWARE_CLASSES = (  
    'django.middleware.common.CommonMiddleware',  
    'django.contrib.sessions.middleware.SessionMiddleware',  
    'django.middleware.csrf.CsrfViewMiddleware',  
    'django.contrib.auth.middleware.AuthenticationMiddleware',  
    'django.contrib.messages.middleware.MessageMiddleware',  
)
```

in settings.py

Django middleware



```
INSTALLED_APPS = (  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.sites',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
)
```

in settings.py

Django installed apps





2. Know your ORM

Release-o-tron Django

localhost:8000

Admin Releases

Application Releases what and when

Id	App	Released
10	Admin	Oct. 29, 2012
9	Frontend	Oct. 29, 2012
8	Frontend	Oct. 29, 2012
7	Admin	Oct. 29, 2012
6	API	Oct. 29, 2012
5	Frontend	Oct. 29, 2012
4	API	Oct. 29, 2012
3	API	Oct. 29, 2012
2	Frontend	Oct. 29, 2012
1	Frontend	Oct. 29, 2012

POWERED BY

Django `select_related()`


```
queryset = Release.objects.all()
```

11 SQL Queries

N+1 problem

The Django logo, consisting of the word "django" in white lowercase letters on a dark green rectangular background.

```
queryset = Release.objects.select_related()
```

1 SQL Query

Joins to the rescue

The Django logo, consisting of the word "django" in a white, lowercase, sans-serif font, set against a dark green rectangular background.

Release-o-tron Rails

localhost:3000/releases

SQL WARNING

Releases Applications

Application Releases

what and when [New release](#)

Id	App	Released
51848956	frontend	3 Nov 2012 17:11PM
113629430	admin	3 Nov 2012 17:11PM
281110143	frontend	3 Nov 2012 17:11PM
298486374	api	3 Nov 2012 17:11PM
374648174	api	3 Nov 2012 17:11PM
507103779	admin	3 Nov 2012 17:11PM
638205575	frontend	3 Nov 2012 17:11PM
980190962	frontend	3 Nov 2012 17:11PM
991788158	api	3 Nov 2012 17:11PM
1018350795	frontend	3 Nov 2012 17:11PM

Active Record Include



Release.all

11 SQL Queries

N+1 problem



```
Release.includes(:app).all
```

1 SQL Query

Joins to the rescue



```
<link href="/assets/application.css?body=1" ...  
<link href="/assets/apps.css?body=1" ...  
<link href="/assets/bootstrap_and_overrides.css?body=1" ...  
<link href="/assets/releases.css?body=1" ...  
<link href="/assets/sample.css?body=1" ...  
<link href="/assets/scaffolds.css?body=1" ...
```

to this

```
<link href="/assets/application.css" ...
```

3. Asset compilation in HTML

```
h1 {
  padding-top: 40px;
}
/*!
 * Bootstrap v2.2.1
 *
 * Copyright 2012 Twitter, Inc
 * Licensed under the Apache License v2.0
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Designed and built with all the love in the world @twitter by @mdo and
@fat.
 */
article, aside, details, figcaption, figure, footer, header, hgroup, nav,
section {
  display: block;
}

to this

h1{padding-top:40px}
article,aside,details,figcaption,figure,footer,header,hgroup,nav,section{display:block}
```

Asset compilation in CSS

```
# Enable the asset pipeline
config.assets.enabled = true

# Version of your assets
config.assets.version = '1.0'

in config/application.rb

# compress assets
config.assets.compress = true

# Don't expands the lines which load the assets
config.assets.debug = false

in config/environments/production.rb
```

Asset compilation configuration





4. Different runtimes





Different runtimes - JRuby





Different runtimes - PyPy

django

```
require 'test_helper'
require 'rails/performance_test_help'

class ReleaseTest < ActionDispatch::PerformanceTest
  self.profile_options = { :runs => 10,
                          :metrics => [:wall_time] }

  def test_release_index
    get '/releases'
  end
end

in test/performance/releases_test.rb
```

5. Performance tests



```
rake test:benchmark
```

```
Started
```

```
ReleaseTest#test_release_index (110 ms warmup)
```

```
    wall_time: 6 ms
```

```
Finished in 0.551815 seconds.
```

```
1 tests, 0 assertions, 0 failures, 0 errors, 0 skips
```

Rails benchmarker



Cache
(everything)



1. Built-in caching support

The screenshot shows a web browser window with the URL `https://docs.djangoproject.com/en/dev/topics/cache/?from=olddocs`. The page title is "Django's cache framework". The navigation bar includes links for Home, Download, Documentation, Weblog, Community, and Code. The main content area starts with the Django logo and the text "Django documentation". The article title is "Django's cache framework". The text explains that dynamic web sites are expensive to process and that caching is essential for high-traffic sites. It includes a code block with pseudocode for caching logic. On the right side, there is a search bar, a version selector set to "Development trunk", and a table of contents with links to various caching topics like Memcached, Database caching, and Filesystem caching.

django

Home Download Documentation Weblog Community Code

Django documentation

Django's cache framework

A fundamental trade-off in dynamic Web sites is, well, they're dynamic. Each time a user requests a page, the Web server makes all sorts of calculations – from database queries to template rendering to business logic – to create the page that your site's visitor sees. This is a lot more expensive, from a processing-overhead perspective, than your standard read-a-file-off-the-filesystem server arrangement.

For most Web applications, this overhead isn't a big deal. Most Web applications aren't washingtonpost.com or slashdot.org; they're simply small- to medium-sized sites with so-so traffic. But for medium- to high-traffic sites, it's essential to cut as much overhead as possible.

That's where caching comes in.

To cache something is to save the result of an expensive calculation so that you don't have to perform the calculation next time. Here's some pseudocode explaining how this would work for a dynamically generated Web page:

```
given a URL, try finding that page in the cache
if the page is in the cache:
    return the cached page
else:
    generate the page
    save the generated page in the cache (for next time)
    return the generated page
```

Search

Version: Development trunk

Search

Contents

- Django's cache framework
 - Setting up the cache
 - Memcached
 - Database caching
 - Database caching and multiple databases
 - Filesystem caching
 - Local-memory caching
 - Dummy caching (for development)
 - Using a custom cache backend

Great documentation 1


...caches that you don't directly control but to which you can provide hints (via HTTP headers) about which parts of your site should be cached, and how.

- The low-level
 - Accessing the cache
- Documentation version: dev

Ruby on Rails Guides: Caching

guides.rubyonrails.org/caching_with_rails.html

More at rubyonrails.org: [Overview](#) | [Download](#) | [Deploy](#) | [Code](#) | [Screencasts](#) | [Documentation](#) | [Ecosystem](#) | [Community](#) | [Blog](#)



RailsGuides

Home **Guides Index** ▾ Contribute Credits

Caching with Rails: An overview

This guide will teach you what you need to know about avoiding that expensive round-trip to your database and returning what you need to return to the web clients in the shortest time possible.

After reading this guide, you should be able to use and configure:

- ✔ Page, action, and fragment caching
- ✔ Sweepers
- ✔ Alternative cache stores
- ✔ Conditional GET support

Chapters

- Basic Caching**
 - [Page Caching](#)
 - [Action Caching](#)
 - [Fragment Caching](#)
 - [Sweepers](#)
 - [SQL Caching](#)
- Cache Stores**
 - [Configuration](#)
 - [ActiveSupport::Cache::Store](#)
 - [ActiveSupport::Cache::MemoryStore](#)

Great documentation 2



without the use of any third party plugins.

To start playing with caching you'll want to ensure that

- [Custom Cache Stores](#)
- [Cache Keys](#)



2. ORM Caching



```
CACHES = {  
    'default' : dict(  
        BACKEND = 'johnny.backends.memcached.MemcachedCache',  
        LOCATION = ['127.0.0.1:11211'],  
        JOHNNY_CACHE = True,  
    )  
}
```

```
MIDDLEWARE_CLASSES = (  
    'johnny.middleware.LocalStoreClearMiddleware',  
    'johnny.middleware.QueryCacheMiddleware',  
) + MIDDLEWARE_CLASSES
```

```
in test/performance/releases_test.rb
```

Johnny Cache



Release-o-tron Django

localhost:8000

Admin Releases

Application Releases what and when

Id	App	Released
10	Admin	Oct. 29, 2012
9	Frontend	Oct. 29, 2012
8	Frontend	Oct. 29, 2012
7	Admin	Oct. 29, 2012
6	API	Oct. 29, 2012
5	Frontend	Oct. 29, 2012
4	API	Oct. 29, 2012
3	API	Oct. 29, 2012
2	Frontend	Oct. 29, 2012
1	Frontend	Oct. 29, 2012

POWERED BY

Hide »

- Versions
DJANGO 1.4.2
- Time
CPU: 360.00ms (367.43ms)
- Settings
- HTTP Headers
- Request Vars
- Templates
- SQL
4 QUERIES IN 2.31ms
- Signals
13 RECEIVERS FROM 12 SIGNALS
- Logging
0 MESSAGES
- Memcache
31 CALLS IN 6.97ms

First request **4 Queries**

django

Release-o-tron Django
localhost:8000

Admin Releases

Application Releases what and when

Id	App	Released
10	Admin	Oct. 29, 2012
9	Frontend	Oct. 29, 2012
8	Frontend	Oct. 29, 2012
7	Admin	Oct. 29, 2012
6	API	Oct. 29, 2012
5	Frontend	Oct. 29, 2012
4	API	Oct. 29, 2012
3	API	Oct. 29, 2012
2	Frontend	Oct. 29, 2012
1	Frontend	Oct. 29, 2012

POWERED BY

Hide »

- Versions
DJANGO 1.4.2
- Time
CPU: 260.00ms (2559.47ms)
- Settings
- HTTP Headers
- Request Vars
- Templates
- SQL
0 QUERIES IN 0.00ms
- Signals
13 RECEIVERS FROM 12 SIGNALS
- Logging
0 MESSAGES
- Memcache
25 CALLS IN 6.88ms

Subsequent requests **0 Queries**



```
class Release < ActiveRecord::Base
  acts_as_cached
  after_save :expire_cache
  attr_accessible :app_id
  belongs_to :app

  def self.recent
    includes(:app).all
  end
end
```

Cache-fu



```
class Release < ActiveRecord::Base
  acts_as_cached
  after_save :expire_cache
  attr_accessible :app_id
  belongs_to :app

  def self.recent
    includes(:app).all
  end
end
```

Add cache behaviour



```
@releases = Release.cached(:recent)
```

Get cached content



Release-o-tron Rails

localhost:3000/releases

SQL WARNING

Releases Applications

Application Releases

what and when [New release](#)

Id	App	Released
51848956	frontend	3 Nov 2012 17:11PM
113629430	admin	3 Nov 2012 17:11PM
281110143	frontend	3 Nov 2012 17:11PM
298486374	api	3 Nov 2012 17:11PM
374648174	api	3 Nov 2012 17:11PM
507103779	admin	3 Nov 2012 17:11PM
638205575	frontend	3 Nov 2012 17:11PM
980190962	frontend	3 Nov 2012 17:11PM
991788158	api	3 Nov 2012 17:11PM
1018350795	frontend	3 Nov 2012 17:11PM

First request **2 Queries**



Release-o-tron Rails

localhost:3000/releases

SQL OK Home Releases Applications

Application Releases

what and when [New release](#)

Id	App	Released
51848956	frontend	3 Nov 2012 17:11PM
113629430	admin	3 Nov 2012 17:11PM
281110143	frontend	3 Nov 2012 17:11PM
298486374	api	3 Nov 2012 17:11PM
374648174	api	3 Nov 2012 17:11PM
507103779	admin	3 Nov 2012 17:11PM
638205575	frontend	3 Nov 2012 17:11PM
980190962	frontend	3 Nov 2012 17:11PM
991788158	api	3 Nov 2012 17:11PM
1018350795	frontend	3 Nov 2012 17:11PM

Subsequent requests **0 Queries**



:// Caching Tutorial for Web Authors and Webmasters

www.mnot.net/cache_docs/

ABOUT BLOG PHOTOS HOME

CACHING TUTORIAL

for Web Authors and Webmasters

This is an informational document. Although technical in nature, it attempts to make the concepts involved understandable and applicable in real-world situations. Because of this, some aspects of the material are simplified or omitted, for the sake of clarity. If you are interested in the minutia of the subject, please explore the [References and Further Information](#) at the end.

http://www.mnot.net/cache_docs/

1. What's a Web Cache? Why do people use them?
2. Kinds of Web Caches
 1. Browser Caches
 2. Proxy Caches
3. Aren't Web Caches bad for me? Why should I help them?
4. How Web Caches Work
5. How (and how not) to Control Caches
 1. HTML Meta Tags vs. HTTP Headers
 2. Pragma HTTP Headers (and why they don't work)
8. Frequently Asked Questions
9. Implementation Notes - Web Servers

3. HTTP is your friend

```
from django.views.decorators.cache import cache_control
@cache_control(public=True, max_age=3600)
```

HTTP headers



```
class ReleasesController < ApplicationController
  def index
    expires_in 60.minute, :public => true
  end
end
```

in app/controllers/releases_controller.rb

HTTP headers



```
INSTALLED_APPS = INSTALLED_APPS + (  
    'varnishapp',  
)
```

```
VARNISH_MANAGEMENT_ADDRS = (  
    'localhost:6082',  
)
```

```
VARNISH_WATCHED_MODELS = ('app.release',)
```

```
in settings.py
```

django-varnish configuration



The screenshot shows a web browser window with the address bar displaying `localhost:8000/admin/varnish/`. The page title is "Django administration" and the user is logged in as "garethr". The main content area shows the Varnish administration interface for `localhost:6082`. At the top, there is a "Run command" button. The interface is divided into several sections:

- Client connections**
 - Connections accepted
 - Requests received
- Cache utilization**
 - Hits
 - Hits for pass
 - Misses
- Backend connection usage**
 - Successful
 - Recycled
 - Reused
 - Unused
 - Not attempted
 - Failure
 - N worker threads 10
 - N worker threads created 10
 - N worker threads not created
 - N worker threads limited
 - N queued work requests
 - N overflowed work requests
 - N dropped work requests
 - N expired objects
 - N objects on deathrow
 - HTTP header overflows
 - Objects sent with sendfile
 - Objects sent with write
- Shared memory**

A vertical "DjDT" logo is visible on the right side of the page.

django-varnish admin



Instrument

(monitor all the things)


```
MIDDLEWARE_CLASSES = MIDDLEWARE_CLASSES + (  
    'django_statsd.middleware.GraphiteRequestTimingMiddleware',  
    'django_statsd.middleware.GraphiteMiddleware',  
)  
  
STATSD_PATCHES = [  
    'django_statsd.patches.db',  
    'django_statsd.patches.cache',  
)  
  
INSTALLED_APPS = INSTALLED_APPS + (  
    'django_statsd',  
)  
  
from settings.py
```

django-statsd

The Django logo, consisting of the word "django" in a white, lowercase, sans-serif font, set against a dark green rectangular background.

Release-o-tron Django

localhost:8000/#

Statsd

Stat	Timing	Time
cache.memcached.get		10ms
view.sample.views.ReleaseListView.GET		51ms
view.sample.views.GET		51ms
view.GET		51ms

Stat	Total	Count	Values
response.200	1	1	

- Hide »
- Versions
DJANGO 1.4.2
- Time
CPU: 50.00ms (52.64ms)
- Settings
- HTTP Headers
- Request Vars
- Templates
- SQL
0 QUERIES IN 0.00ms
- Signals
13 RECEIVERS FROM 12 SIGNALS
- Logging
0 MESSAGES
- Memcache
5 CALLS IN 1.38ms
- Statsd

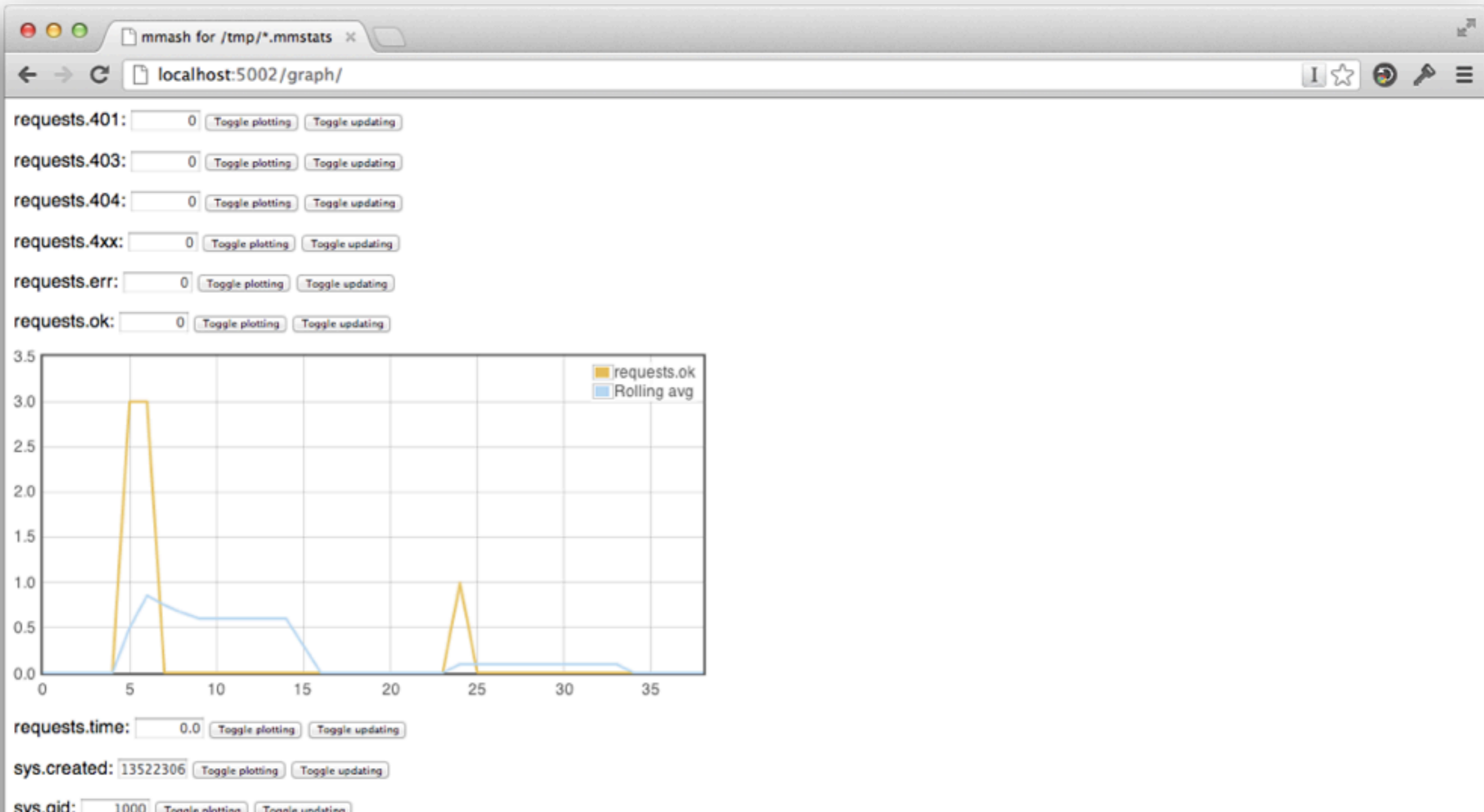
django-statsd on debug toolbar



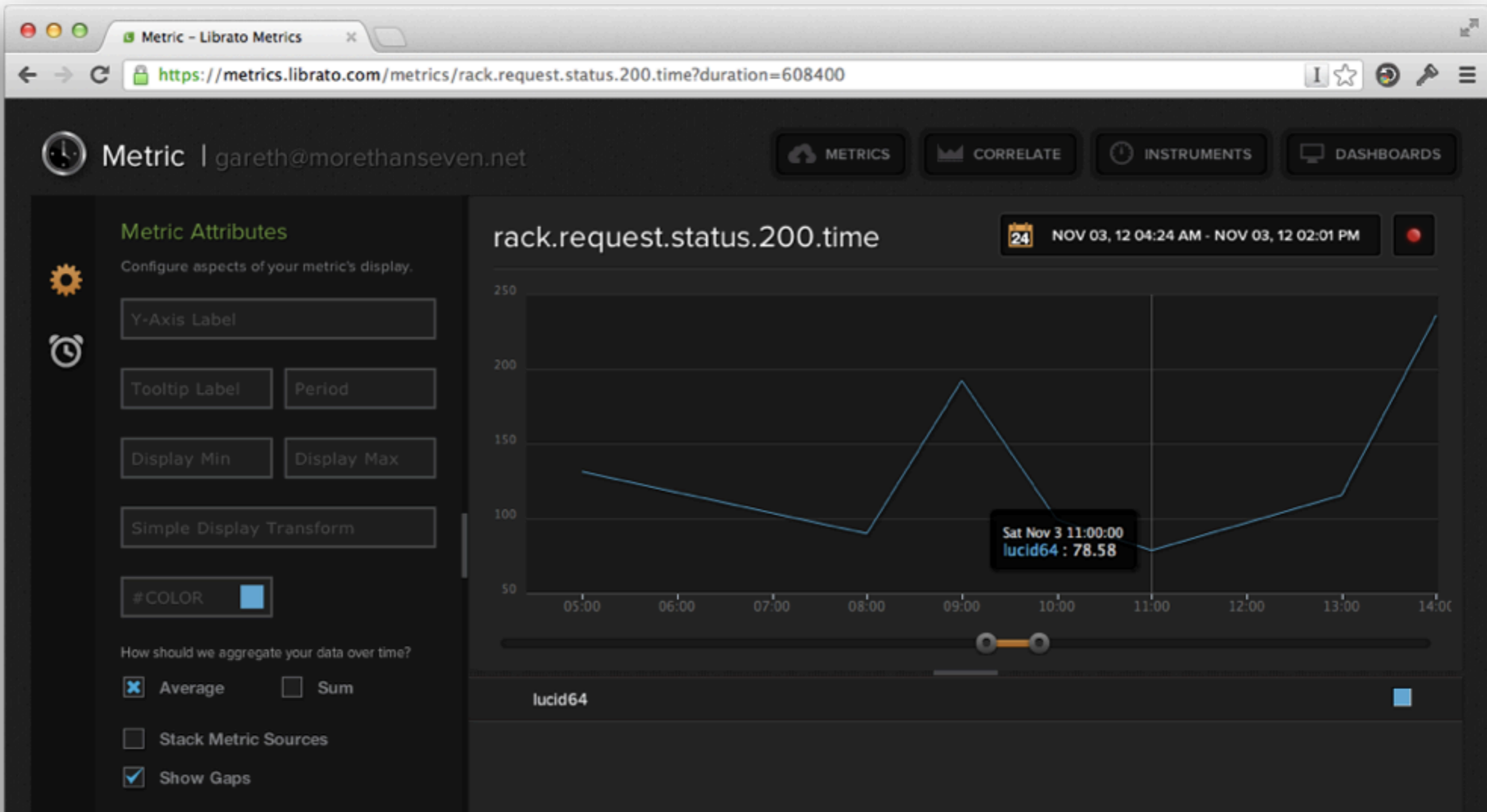
```
MIDDLEWARE_CLASSES = (  
    'django_mmstats.middleware.MmStatsMiddleware',  
)  
  
import stats  
MMSTATS_CLASS = stats.DjangoStats  
  
from settings.py  
  
from django_mmstats.base import BaseDjangoStats  
  
class DjangoStats(BaseDjangoStats):  
    """Add mmstats fields here, just like Django models!"""  
  
from stats.py
```

django-mmstats

The Django logo, consisting of the word "django" in a white, lowercase, sans-serif font, set against a dark green rectangular background.



mmash



librato-rails



**Not just in
development**
(measure in production)




1. Logs (again)

etsy/logster · GitHub




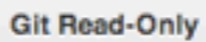
GitHub, Inc. [US] <https://github.com/etsy/logster>


github [Signup and Pricing](#) [Explore GitHub](#) [Features](#) [Blog](#) [Sign in](#)


PUBLIC  **etsy / logster** ★ Star 600 🍴 Fork 74

Code Network Pull Requests 2 Issues 7 Graphs

Parse log files, generate metrics for Graphite and Ganglia — [Read more](#)


 Clone in Mac
  ZIP
  HTTP
  Git Read-Only




 Read-Only access



 branch: master
 Files Commits Branches 1 Tags 1 Downloads

logster / 📦 60 commits

Merge pull request #25 from chino/add-parsers-to-setup-py

 **mrtazz** authored 3 days ago 📦 latest commit fef5fb9789

 bin	16 days ago	Merge branch 'pythonize-2' of https://github.com/msabramo/logster [mrtazz]
 logster	16 days ago	adapt MetricLogster to packaged format [mrtazz]
 tests	16 days ago	adapt MetricLogster to packaged format [mrtazz]

 README.md	16 days ago	fix minor README formatting bug [mrtazz]
 setup.py	6 days ago	adding logster/parsers to be packaged with the egg [chino]

Logster

GitHub repository page for **roidrage / lograge**. The page shows the repository name, a description: "An attempt to tame Rails' default policy to log everything. — Read more", and a link to a blog post: <http://www.paperplanes.de/2012/3/14/on-notifications-logsubscribers-and-bringing-sanity-to-rails-logging.html>. The repository has 209 stars and 18 forks. The current branch is **master**. The commit history shows:

File	Time	Description
lib	2 months ago	Handle Rails 3.0.x request format payload [urbanautomaton]
spec	6 months ago	Return status=0 when the status is unknown. [roidrage]
.gitignore	8 months ago	Initial and incomplete commit [roidrage]
rspec	8 months ago	Extracted logging code from Travis. [roidrage]
README.md	7 months ago	Add changes section. [roidrage]

Lograge



```
Started GET "/" for 127.0.0.1 at 2012-03-10 14:28:14 +0100
Processing by HomeController#index as HTML
  Rendered text template within layouts/application (0.0ms)
  Rendered layouts/_assets.html.erb (2.0ms)
  Rendered layouts/_top.html.erb (2.6ms)
  Rendered layouts/_about.html.erb (0.3ms)
  Rendered layouts/_google_analytics.html.erb (0.4ms)
Completed 200 OK in 79ms (Views: 78.8ms | ActiveRecord: 0.0ms)
```

to this

```
GET /jobs/833552.json format=json action=jobs#show status=200
duration=58.33 view=40.43 db=15.26
```

Lograge





Logstash

timestamp, index them, and search them

All your logs from all over your infrastructure in one place - with
searching and parsing. Give us your logs, we'll handle the rest.

logging: logstash and other things - Jord [Share](#) [More info](#)

```
LOGRAGE %{\WORD:method}%{\SPACE}%{\DATA}%  
{SPACE}action=%{\WORD:controller}#%  
{WORD:action}%{\SPACE}status=%{\INT:status}%  
{SPACE}duration=%{\NUMBER:duration}%{\SPACE}  
view=%{\NUMBER:view} (%{\SPACE}db=%  
{NUMBER:db})?%{\GREEDYDATA}
```

Logstash plus lograge



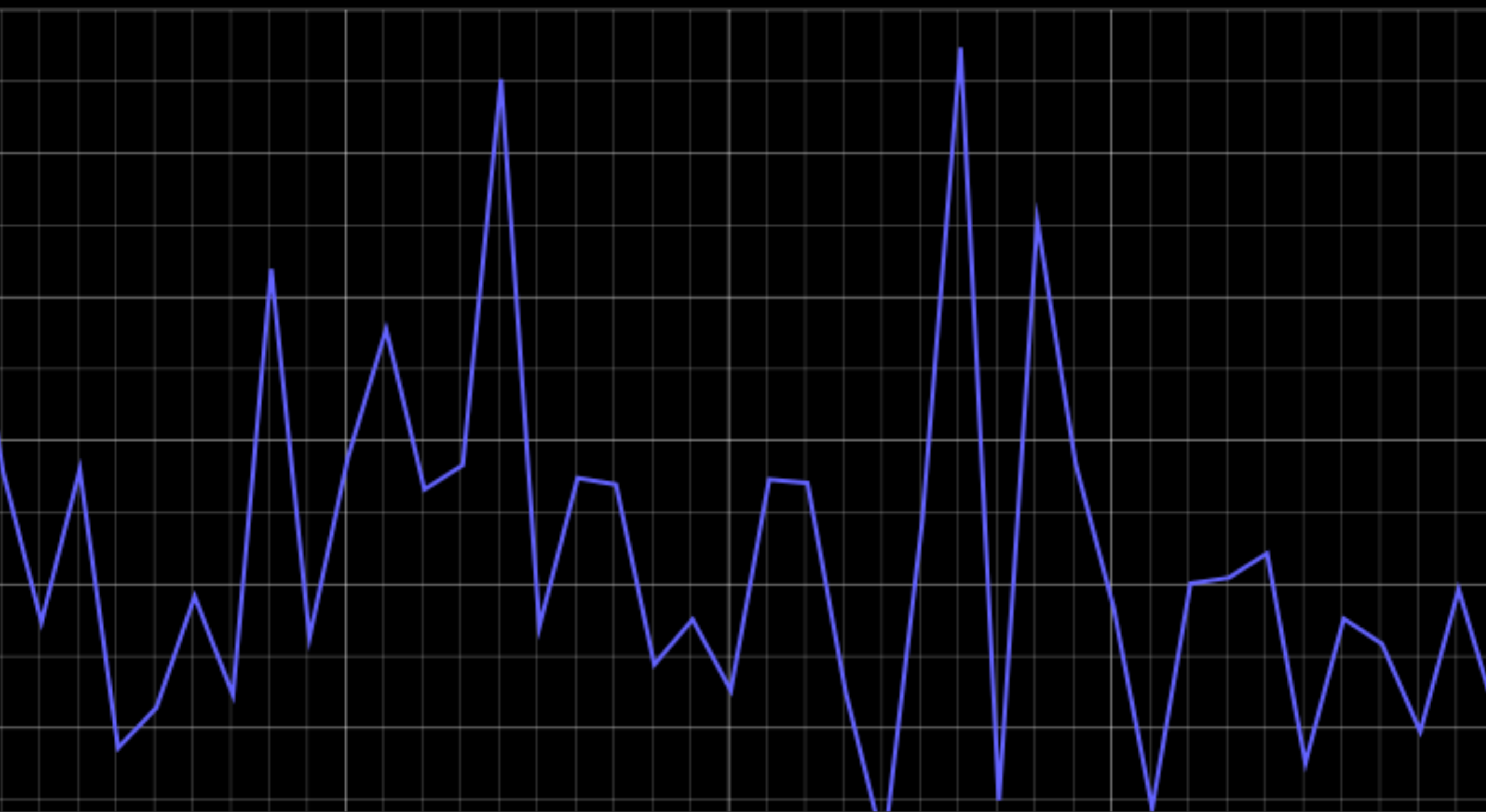
```
LOGRAGE  %{WORD:method}%{SPACE}%{DATA}%  
{SPACE}action=%{WORD:controller}#%  
{WORD:action}%{SPACE}status=%{INT:status}%  
{SPACE}duration=%{NUMBER:duration}%{SPACE}  
view=%{NUMBER:view} (%{SPACE}db=%  
{NUMBER:db}) ?%{GREEDYDATA}
```

Rails performance data



```
output {
  statsd {
    host => "localhost"
    tags => [ "lograge" ]
    timing => [ "<%= @title %>.{controller}.{action}.{method}.duration", "%{duration}" ]
  }
  statsd {
    host => "localhost"
    tags => [ "lograge" ]
    timing => [ "<%= @title %>.{controller}.{action}.{method}.view", "%{view}" ]
  }
}
```

Output to statsd



2. Metrics

18:15

18:20

18:25

18:30

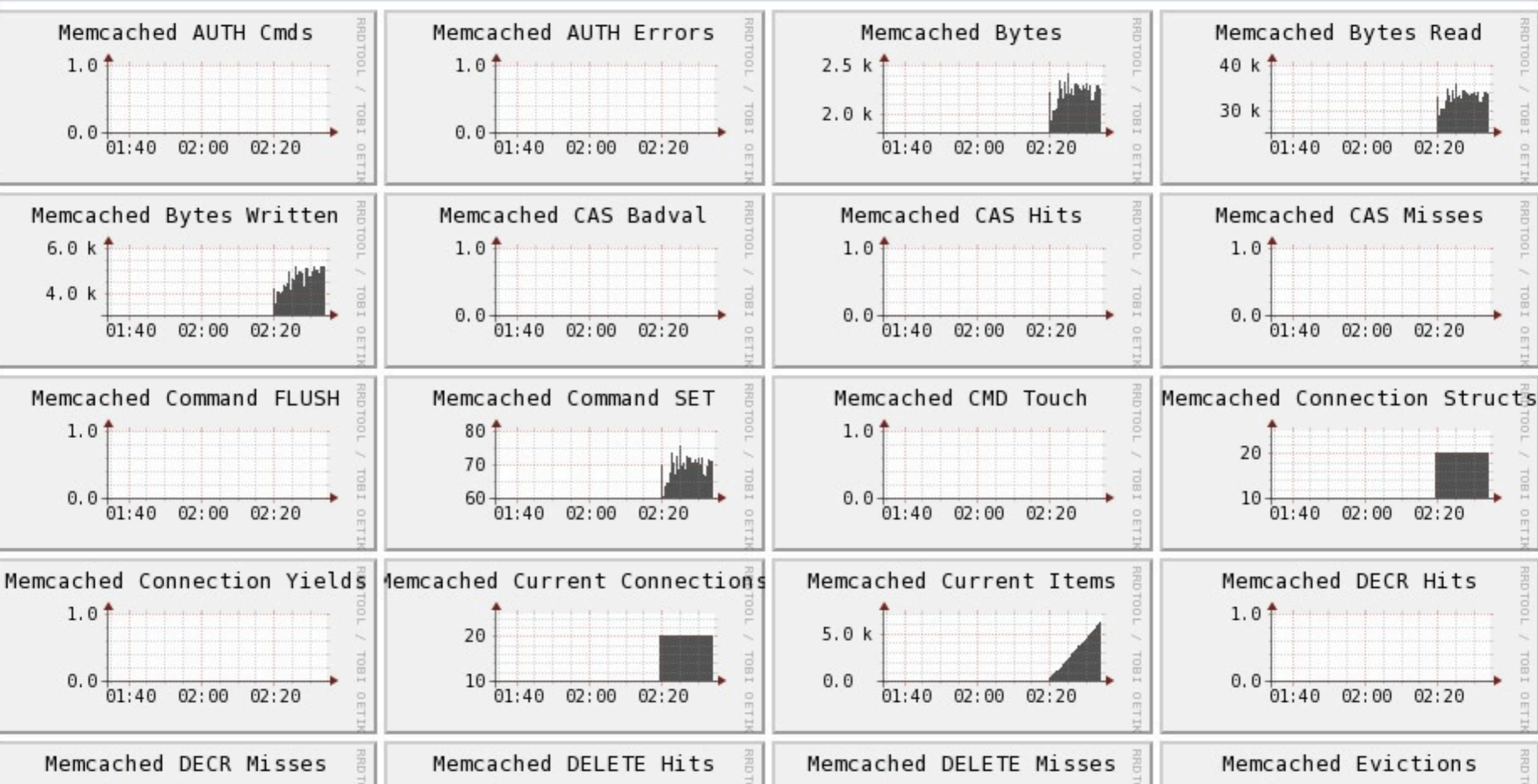
18:35

18:40

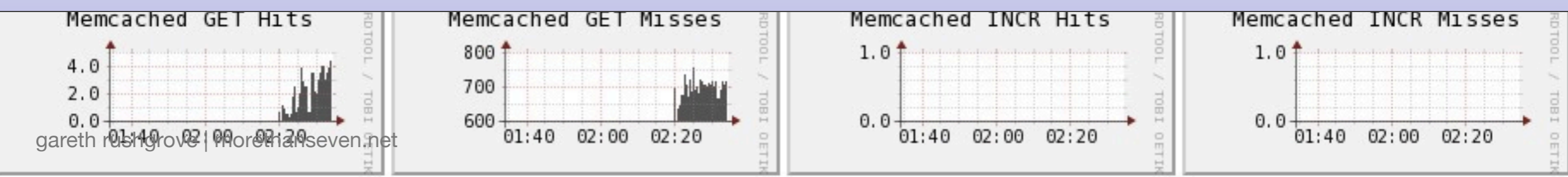
18:45

18:50

memcache metrics (28)



Ganglia

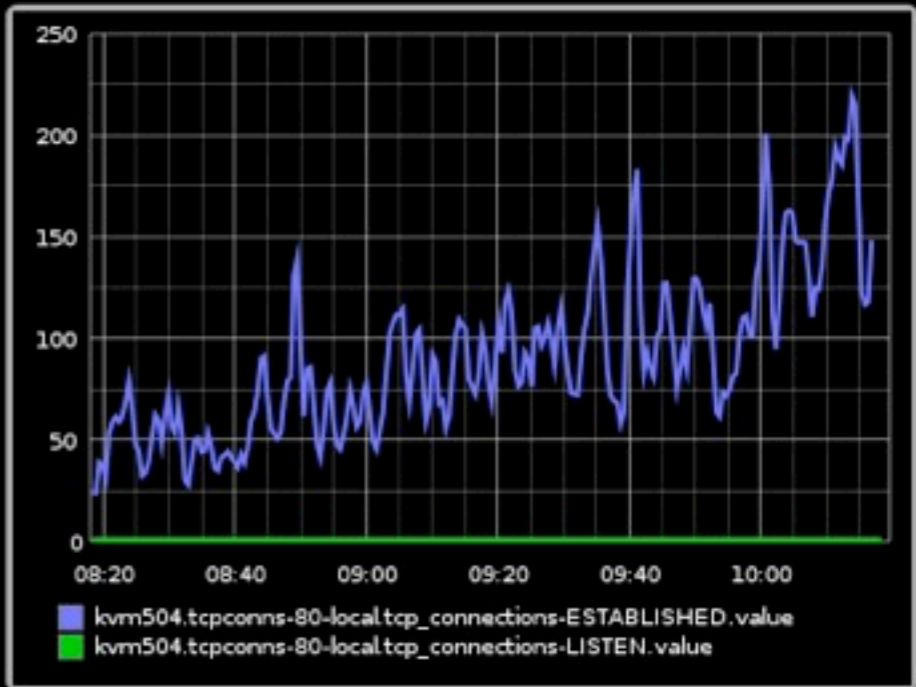
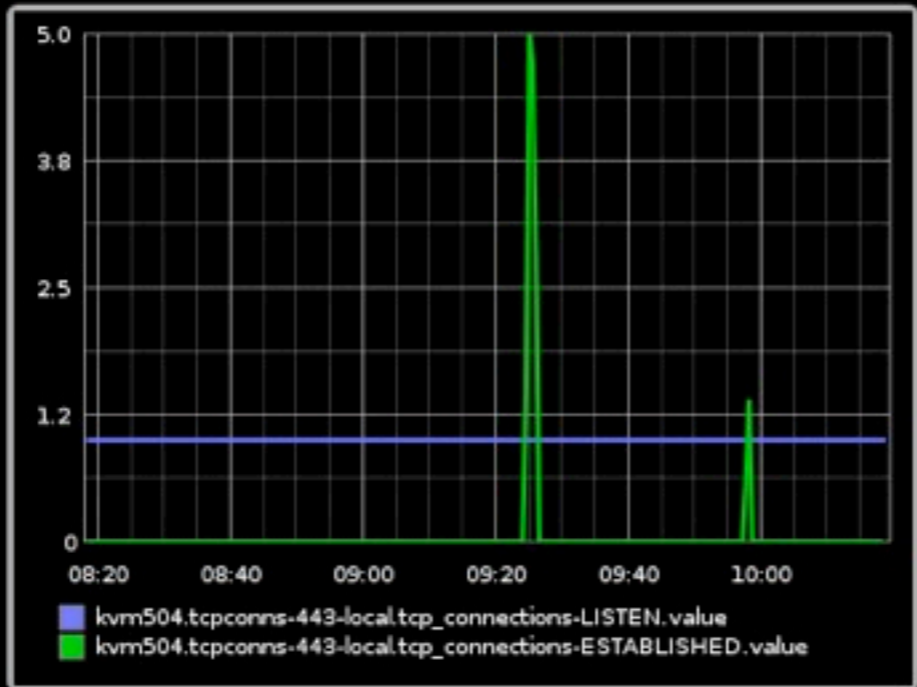
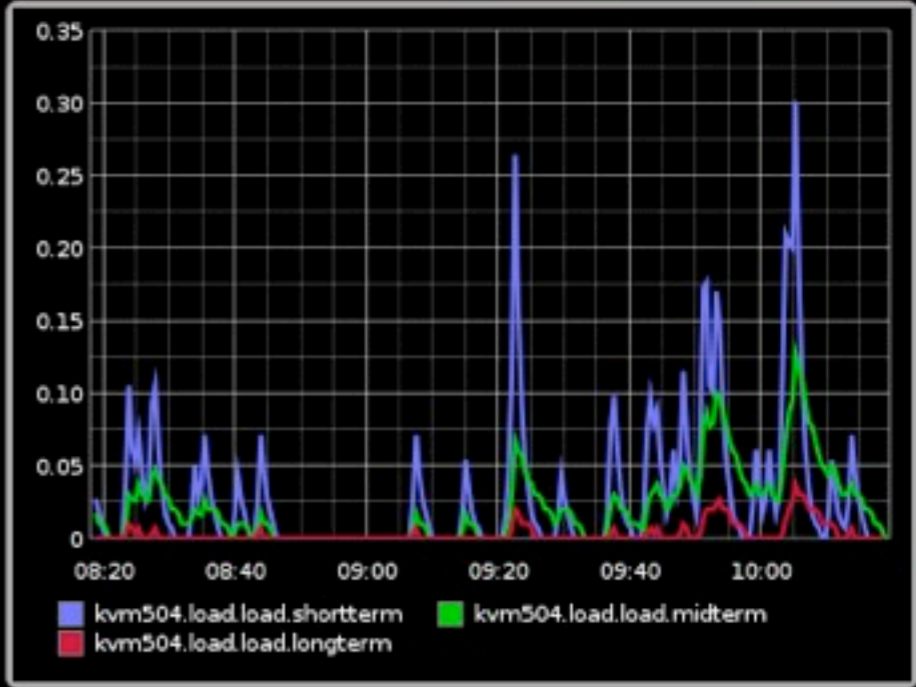
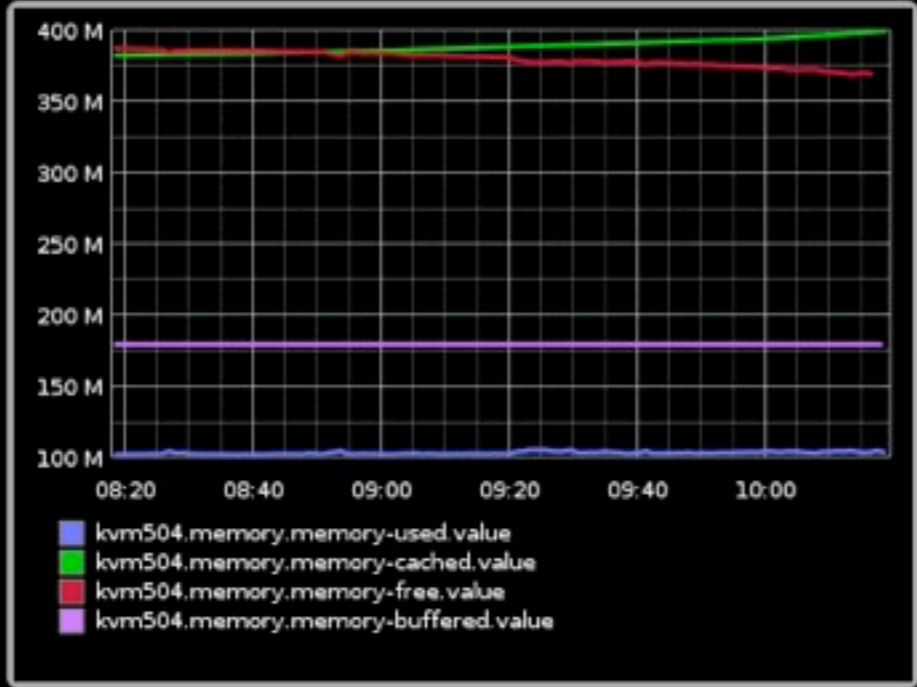


kvm504 - (https://maas.monitoring-fr.org/dashboard/kvm504)

Metric Type: Everything

Category: kvm606

- cpu-0
- df-boot
- df-root
- disk-dm-0
- disk-dm-1
- disk-sr0
- disk-vda
- disk-vda1
- disk-vda2
- disk-vda5
- entropy
- filecount-nginx-etc
- interface-eth0
- load
- memory
- nginx
- processes
- processes-collectd
- processes-nginx
- processes-rsyslog
- protocols-Tcp
- swap
- tail-nginx
- tcpconns-22-local
- tcpconns-27017-local



Graphite

Dashboard (as of 13:45:23)

Sad Pandas

- solr2.tx disk /ssd1
- queue thumbnail_delayed

Health

	cpu	disk /	disk /boot	disk /disk1	disk /disk2	disk /ssd1	disk /ssd2	disk /ssd3	load	memory
api1.tx	0.23	0.53	0.67	0.01	0.82				0.19	0.49
api2.tx	0.36	0.41	0.67	0.01	0.77				0.23	0.52
api3.tx	0.21	0.44	0.67	0.01	0.80				0.21	0.54
api4.tx	0.23	0.42	0.67	0.01	0.79				0.19	0.52
api5.tx	0.21	0.43	0.67	0.01	0.79				0.19	0.52
api7.tx	0.26	0.41	0.67	0.01	0.80				0.19	0.53
be1.tx	0.10	0.03	0.53						0.01	0.27
be2.tx	0.15	0.04	0.53						0.06	0.38
hub3.tx	0.00	0.01	0.43						0.01	0.31
importer.tx	0.11	0.15	0.45		0.01				0.01	0.63
solr2.tx	0.11	0.44	0.67			0.91	0.29	0.40	0.57	0.81

Tablet

videos total	31546476.00
--------------	-------------

Queue

apns	0.00
tablet_batched_user_import	2.00
tablet_featured	0.00
tablet_feed	1405.00
tablet_importer_feed	0.00
tablet_thumbnails	0.00
tablet_user_import	0.00
tablet_user_import_new	0.00
thumbnail_delayed	84075.00
unfollow	0.00
unlike	0.00
user_thumbnails	0.00

API

	0	0.5	0.95	0.99	1	rate
	3.71	36.30	326.78	900.08	900.08	6.32

Importer

	facebook rate	feed creator	tumblr rate	twitter rate	youtube rate
importer.tx	27.57	ok	1.00	88.48	1.00

Reaper

	deletes i	missing ring	scans
be1.tx	458304.00	0.00	383.00
be2.tx	22250070.00	0.00	

APNS

be1.tx	0.00
be2.tx	0.03

Redis

	blocked_clients	connected_clients	db0 keys	memory
be1.tx	0.00	1.00		16.76
be2.tx	20.00	165.00	15.00	4570.46

Thumbnailer

	0	1	2	3	5
be1.tx	2.30				
be2.tx	1.00	ok	ok	ok	ok

User Thumbnailer

be1.tx	0.00
be2.tx	0.00

Featured

Riemann

api2.tx	1.00	25.50	0.04	1.64
be1.tx	1.00	18.40	0.03	5.03
be2.tx	1.00	45.20	0.05	4.76

api4.tx	43641720.00	0.00
api5.tx	40134088.00	1127.59
api7.tx	ok	45567616.00 663.51

User scores

	i	ring	scans
be1.tx	81312.00	0.00	26.99
be2.tx	30240.00	0.00	105.78

Stalker

i	ring	scans
---	------	-------

One-offs

Riak

disk	get 50	get 95	get 99	node_gets	node_puts	put 50	put 95	put 99	read_repairs	ring	vnode_gets	vnode_puts
------	--------	--------	--------	-----------	-----------	--------	--------	--------	--------------	------	------------	------------

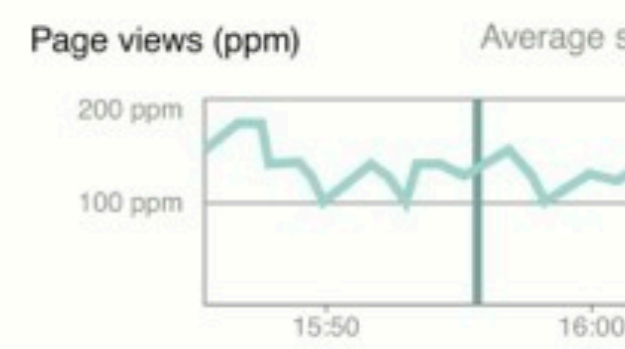
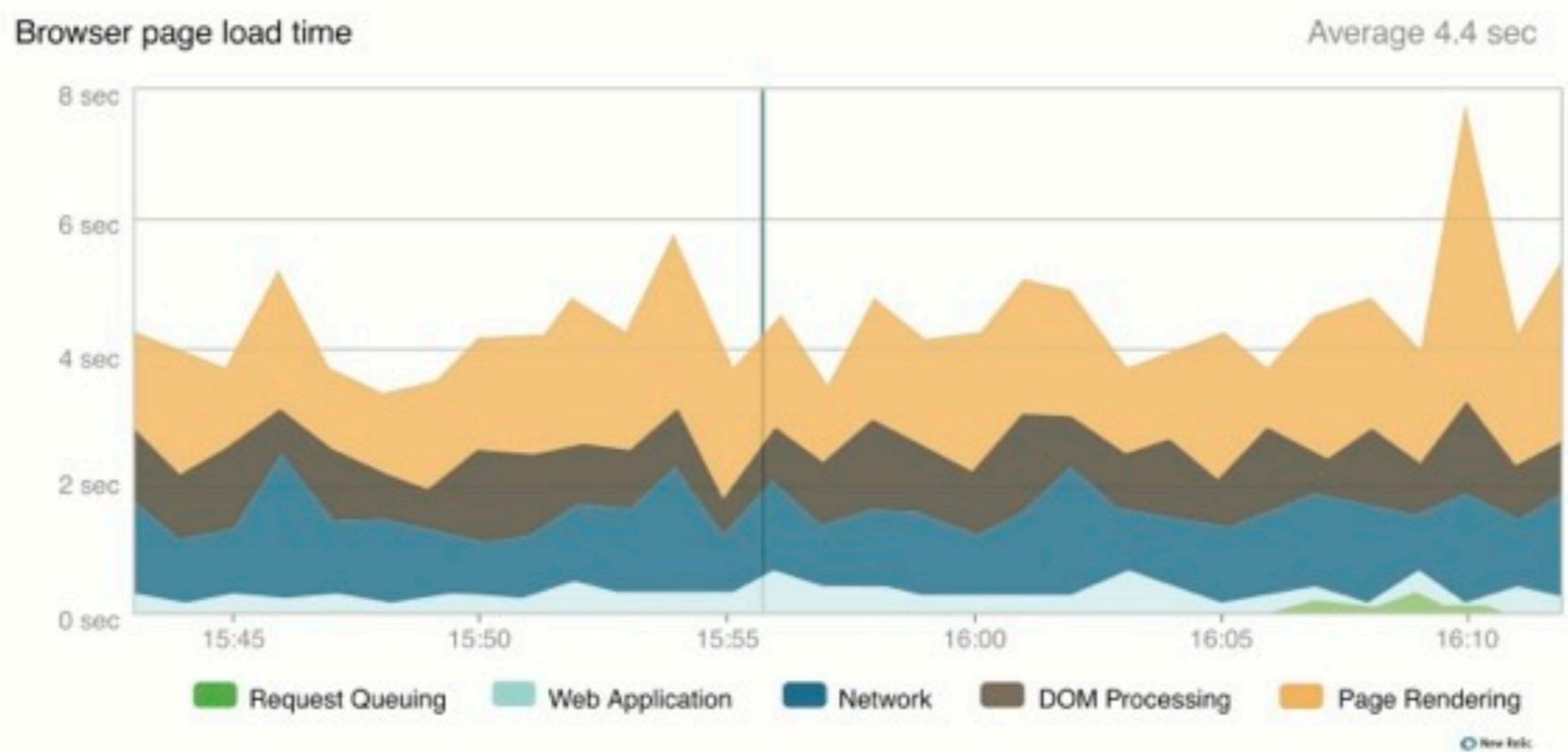
Your Company Last 12 hours

Status 2.51 s 2,283 ppm 153 ms

- APPLICATIONS
 - End user
 - App server
 - Diagnostics
 - Reports
 - Settings

- INCIDENTS
- NOTES
- EMBEDDED CHARTS
- CUSTOM VIEWS

Overview Web pages Browsers Global United States



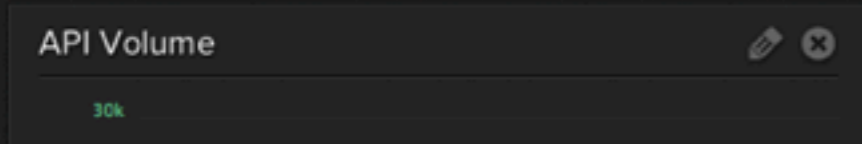
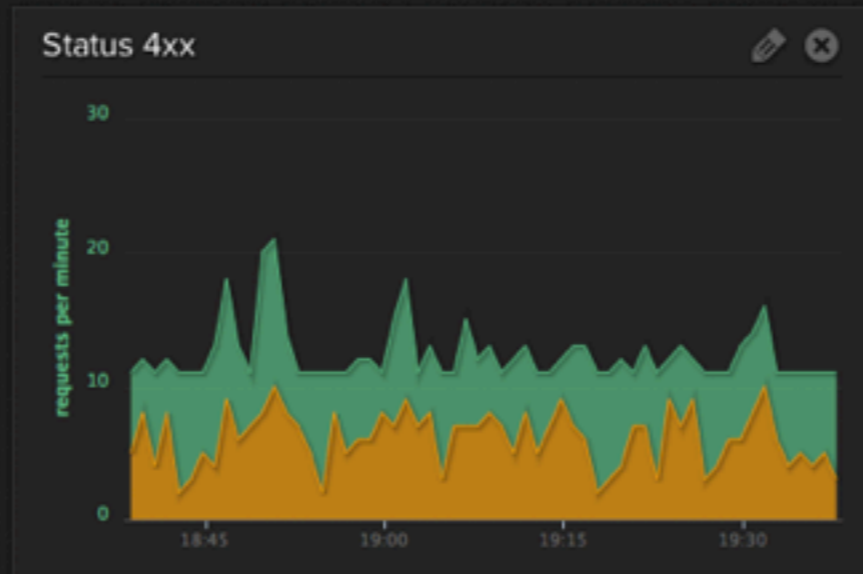
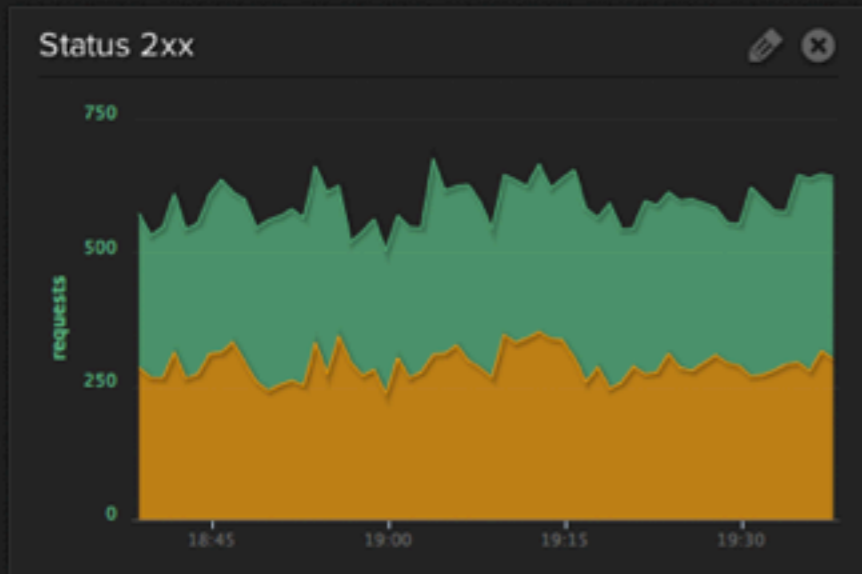
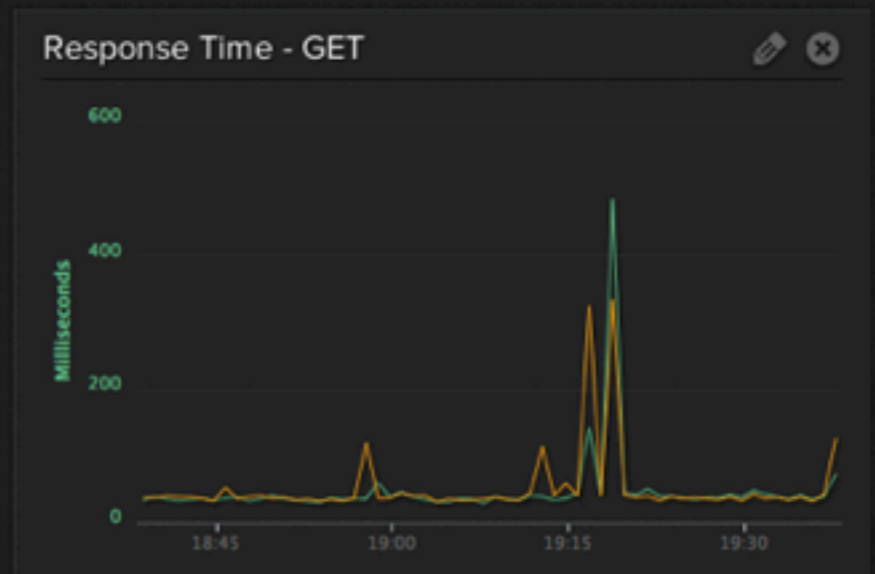
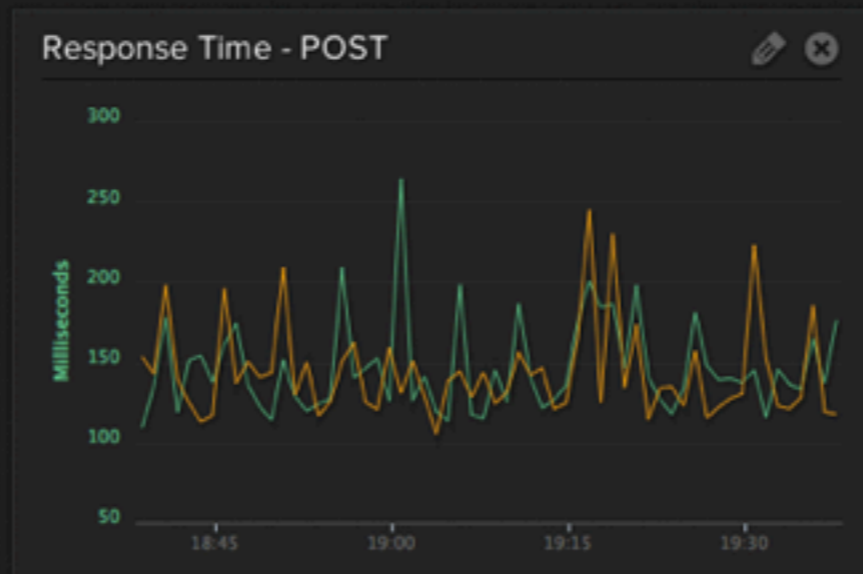
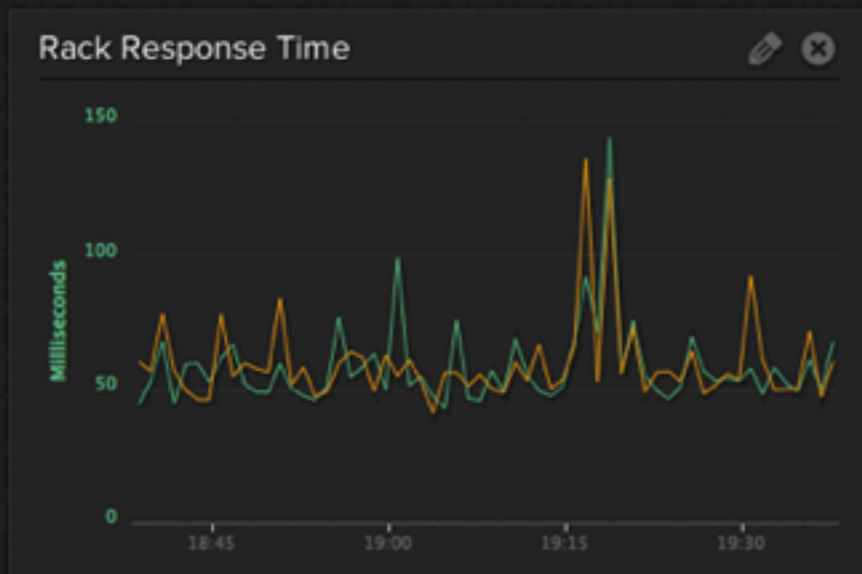
Recent events

All

New Relic newrelic.com

United States United Kingdom Canada Australia

Today, 6:50:34 jesstom deployed revision bf9302784c983187192



Librato Metrics metrics.librato.com

Conclusions

(if all you remember is)



Tooling helps. A lot.

- A debug **toolbar**
- Transparent **caching** support
- Hooks for **instrumentation**
- Configurable **logging**

Your framework should have



If not... build them



Share everything

The End



www.flickr.com/photos/snugglepup/

Thanks for the amazing photos

TALK NERDY TO ME
©2002 Unamerican Activities • "this nation's saving grace" • <http://www.unamerican.com>

Questions?