

# PYTHON PROFILING

PYCON ■ 2013



<http://speakerdeck.com/amjith/>

@amjithr

WHAT'S

PROFILING?



@amjithr

WHAT'S

# PROFILING?

Measuring Performance



@amjithr

Hi



@amjithr

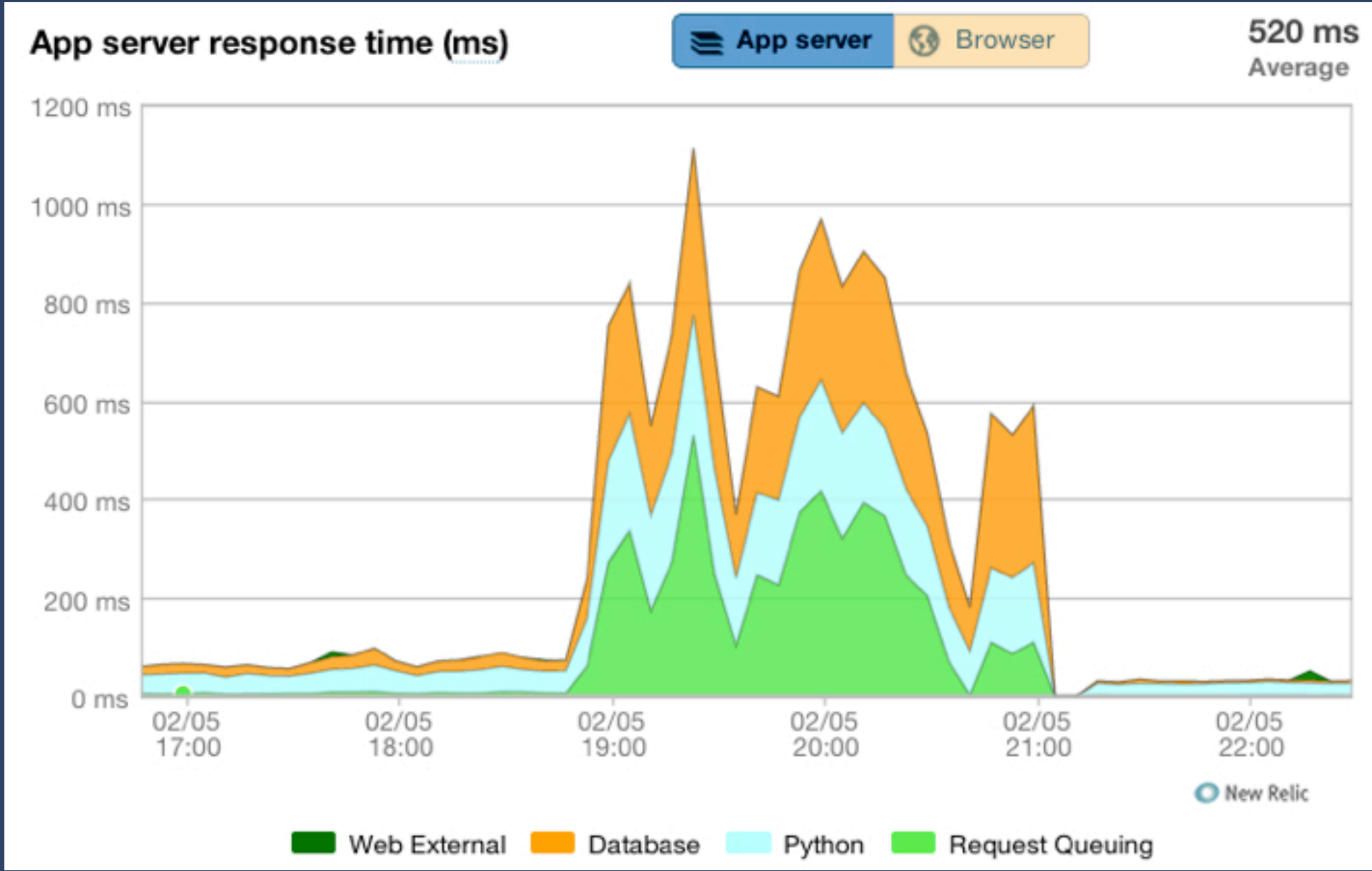




@amjithr



@amjithr



# PERFORMANCE!



@amjithr

# cPROFILE





```
$ python -m cProfile lcm.py
```

```
7780242 function calls in 3.822 seconds
```

```
Ordered by: standard name
```

| ncalls  | tottime | percall | cumtime | percall | filename:lineno(function) |
|---------|---------|---------|---------|---------|---------------------------|
| 1       | 0.000   | 0.000   | 3.822   | 3.822   | lcm.py:12(<module>)       |
| 1       | 2.383   | 2.383   | 3.822   | 3.822   | lcm.py:12(lcm)            |
| 3890120 | 0.724   | 0.000   | 0.724   | 0.000   | {max}                     |
| 1       | 0.000   | 0.000   | 0.000   | 0.000   | {method 'disable' of      |
| 3890119 | 0.716   | 0.000   | 0.716   | 0.000   | {min}                     |



```
$ python -m cProfile lcm.py
```

```
5 function calls in 0.811 seconds
```

```
Ordered by: standard name
```

| ncalls | tottime | percall | cumtime | percall | filename:lineno(function) |
|--------|---------|---------|---------|---------|---------------------------|
| 1      | 0.000   | 0.000   | 0.811   | 0.811   | lcm.py:2(<module>)        |
| 1      | 0.811   | 0.811   | 0.811   | 0.811   | lcm.py:2(lcm)             |
| 1      | 0.000   | 0.000   | 0.000   | 0.000   | {max}                     |
| 1      | 0.000   | 0.000   | 0.000   | 0.000   | {method 'disable' of      |
| 1      | 0.000   | 0.000   | 0.000   | 0.000   | {min}                     |



```
$ python -m cProfile lcm.py
```

```
5 function calls in 0.811 seconds
```

```
Ordered by: standard name
```

| ncalls | tottime | percall | cumtime | percall | filename:lineno(function) |
|--------|---------|---------|---------|---------|---------------------------|
| 1      | 0.000   | 0.000   | 0.811   | 0.811   | lcm.py:2(<module>)        |
| 1      | 0.811   | 0.811   | 0.811   | 0.811   | lcm.py:2(lcm)             |
| 1      | 0.000   | 0.000   | 0.000   | 0.000   | {max}                     |
| 1      | 0.000   | 0.000   | 0.000   | 0.000   | {method 'disable' of      |
| 1      | 0.000   | 0.000   | 0.000   | 0.000   | {min}                     |

# TADA!





# I KNOW PROFILING



@amjithr

```
$ python -m cProfile shorten.py
```

```
88466 function calls (86374 primitive calls) in 0.220 seconds
```

```
Ordered by: standard name
```

| ncalls | tottime | percall | cumtime | percall | filename:lineno(function)           |
|--------|---------|---------|---------|---------|-------------------------------------|
| 1      | 0.000   | 0.000   | 0.000   | 0.000   | <string>:1(<lambda>)                |
| 1      | 0.000   | 0.000   | 0.000   | 0.000   | <string>:1(<module>)                |
| 1      | 0.000   | 0.000   | 0.000   | 0.000   | <string>:1(AnnotatedCTE)            |
| 1      | 0.000   | 0.000   | 0.000   | 0.000   | <string>:1(AnnotatedCase)           |
| 1      | 0.000   | 0.000   | 0.000   | 0.000   | <string>:1(AnnotatedCast)           |
| 1      | 0.000   | 0.000   | 0.000   | 0.000   | <string>:1(AnnotatedClauseElement)  |
| 1      | 0.000   | 0.000   | 0.000   | 0.000   | <string>:1(AnnotatedClauseList)     |
| 1      | 0.000   | 0.000   | 0.000   | 0.000   | <string>:1(AnnotatedColumn)         |
| 1      | 0.000   | 0.000   | 0.000   | 0.000   | <string>:1(AnnotatedColumnClause)   |
| 1      | 0.000   | 0.000   | 0.000   | 0.000   | <string>:1(AnnotatedColumnElement)  |
| 1      | 0.000   | 0.000   | 0.000   | 0.000   | <string>:1(AnnotatedCompoundSelect) |
| 1      | 0.000   | 0.000   | 0.000   | 0.000   | <string>:1(AnnotatedDelete)         |
| .....  | .....   | .....   | .....   | .....   | .....                               |
| .....  | .....   | .....   | .....   | .....   | .....                               |



# PROFILE ANALYSIS

SAVE & ANALYZE



@amjithr

# SAVE

```
$ python -m cProfile -o file.prof script.py
```



# SAVE

```
$ python -m cProfile -o file.prof script.py
```

# ANALYZE

```
>>> import pstats
>>> p = pstats.Stats('shorten.prof')
>>> p.sort_stats('calls')
<pstats.Stats instance at 0x105ac8248>
>>> p.print_stats(5)
List reduced from 1979 to 5 due to restriction <5>
ncalls  tottime  percall  cumtime  percallfilename:lineno(function)
 8053    0.007    0.000    0.007    0.000 {isinstance}
 2794    0.004    0.000    0.004    0.000 {built-in method match}
```



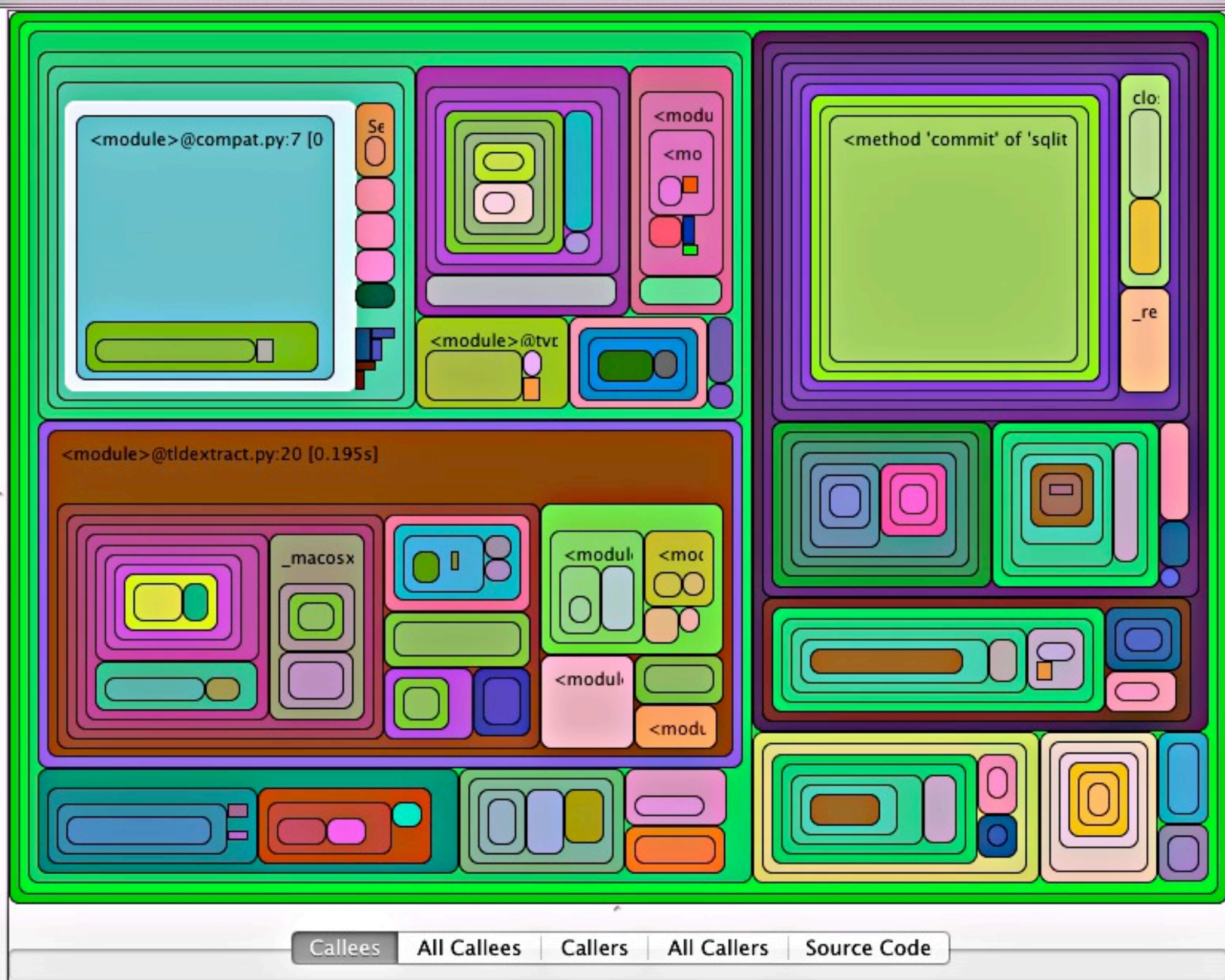


# RunSnakeRun



@amjithr

| Name         | Calls | RCalls |
|--------------|-------|--------|
| <module>     | 0     | 2      |
| <module>     | 1     | 1      |
| <module>     | 1     | 1      |
| visit        | 13    | 13     |
| <module>     | 1     | 1      |
| increment... | 13    | 13     |
| <module>     | 1     | 1      |
| <module>     | 1     | 1      |
| commit       | 26    | 26     |
| commit       | 13    | 13     |
| commit       | 13    | 13     |
| _do_commit   | 13    | 13     |
| _commit_i... | 13    | 13     |
| do_commit    | 13    | 13     |
| <method ...  | 13    | 13     |
| <module>     | 1     | 1      |
| <module>     | 1     | 1      |
| <module>     | 1     | 1      |
| _compile     | 149   | 149    |
| compile      | 62    | 62     |
| compile      | 56    | 56     |
| <module>     | 1     | 1      |
| <module>     | 1     | 1      |
| first        | 39    | 39     |
| __getitem__  | 39    | 39     |
| Environm...  | 1     | 1      |
| get_suppo... | 1     | 1      |
| __iter__     | 41    | 41     |
| execute      | 55    | 55     |
| _execute_... | 54    | 54     |
| execute      | 41    | 41     |



Callees All Callees Callers All Callers Source Code

<module>@shorten.py:1 [0.763s]

**THAT IS  
PERFECTION!**



@amjithr

**THAT IS  
PERFECTION!**

**JUST ONE CATCH.**



@amjithr

```
$ time python fib.py
```

```
python fib.py 5.43s user 0.01s system 99% cpu 5.451 total
```

```
$ time python -m cProfile fib.py
```

```
python -m cProfile fib.py 9.63s user 0.04s system 99% cpu 9.670 total
```



```
$ time python fib.py
```

```
python fib.py 5.43s user 0.01s system 99% cpu 5.451 total
```

```
$ time python -m cProfile fib.py
```

```
python -m cProfile fib.py 9.63s user 0.04s system 99% cpu 9.670 total
```

~2x OVERHEAD





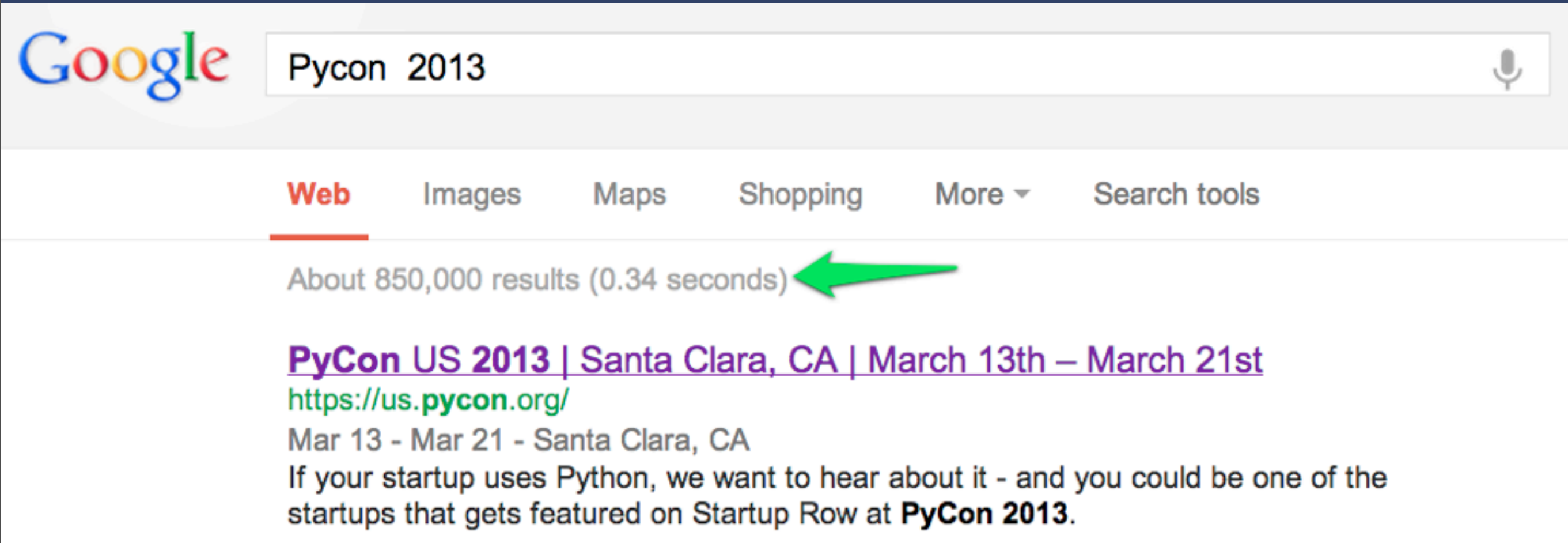
You can't run it in production



@amjithr




# PROFILER IN PRODUCTION



Google

**Web** Images Maps Shopping More ▾ Search tools

About 850,000 results (0.34 seconds) 

**[PyCon US 2013 | Santa Clara, CA | March 13th – March 21st](https://us.pycon.org/)**  
<https://us.pycon.org/>  
Mar 13 - Mar 21 - Santa Clara, CA  
If your startup uses Python, we want to hear about it - and you could be one of the startups that gets featured on Startup Row at **PyCon 2013**.





# cProfile

SLOW

~~PRODUCTION~~

NOT REALISTIC



# TARGETED PROFILING

## CRITICAL FUNCTIONS



```
from profile_func import profile_func

@profile_func
def lcm(arg1, arg2):
    mx = i = max(arg1, arg2)
    mn = min(arg1, arg2)
    while i < (arg1 * arg2):
        if i % mn == 0:
            return i
        i += mx
    return(arg1 * arg2)
```



# TARGETED PROFILING

PROs

Less overhead

Finer Resolution

CONs

Still slow

Manual



# NEWRELIC

TARGETED  
HYBRID PROFILING  
WEB APPS ONLY



@amjithr

# TARGET



WEB FRAMEWORKS  
VIEW HANDLERS  
SQL CALLS



# HYBRID PROFILING

TOP-LEVEL FUNCTION ONLY  
TIMING DATA  
CAPTURE ARGUMENTS



# TRANSACTION TRACE

Expand performance problems Collapse all

| Duration (ms) | Duration (%)  | Segment   | Drilldown | Timestamp |
|---------------|---------------|---|-----------|-----------|
| 358           | 100.00%       | /renamed-basic.blog.views:post_detail                   |           | 0.000 s   |
| 356           | 99.44%        | ▼ WSGI/Application                                      |           | 0.000 s   |
| 356           | 99.44%        | ▼ django.core.handlers.wsgi:WSGIHandler.__call__        |           | 0.000 s   |
| 2.0           | 0.56%         | ▼ 6 fast method calls                                   |           | 0.002 s   |
| <b>93.0</b>   | <b>25.98%</b> | ▼ basic.blog.views:post_detail                          |           | 0.004 s   |
| 4.0           | 1.12%         | sqlite3.dbapi2:connect                                  |           | 0.005 s   |
| 17.0          | 4.75%         | blog_posts - SELECT                                     |           | 0.010 s   |
| 11.0          | 3.07%         | ▶ 6 fast method calls                                   |           | 0.038 s   |
| 48.0          | 13.41%        | ▶ Render/blog/post_detail.html                          |           | 0.049 s   |
| 233           | 65.08%        | ▼ request.middleware:RequestMiddleware.process_response |           | 0.097 s   |
| 0             | 0.00%         | request.models:Request.__init__                         |           | 0.098 s   |
| <b>217</b>    | <b>60.61%</b> | request_request - INSERT                                |           | 0.098 s   |



**SQL query** ✕

```
INSERT INTO "request_request" ("response", "method", "path", "time", "is_secure", "is_ajax", "ip", "user_id", "referer", "user_agent", "language") VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)
```

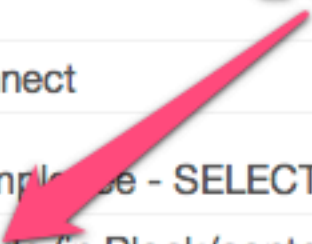
|             |              |  |  |         |
|-------------|--------------|--|--|---------|
| 7.0         | 1.96%        | other - SQL  |  | 0.316 s |
| <b>19.0</b> | <b>5.31%</b> | slimmer.middleware:CompressHtmlMiddleware.process_response |  | 0.330 s |
| 4.0         | 1.12%        | ▶ 7 fast method calls                                      |  | 0.349 s |
| 2.0         | 0.56%        | WSGI/Response  |  | 0.356 s |





| Duration (ms) | Duration (%) | Segment   | Drilldown  | Timestamp |
|---------------|--------------|---|--|-----------|
| 2,710         | 100.00%      | /employees.views:filtering  |  | 0.000 s   |
| 2,710         | 99.89%       | WSGI/Application  |  | 0.000 s   |
| 2,710         | 99.89%       | django.core.handlers.wsgi:WSGIHandler   |  | 0.000 s   |
| 2.0           | 0.07%        | 6 fast method calls   |  | 0.000 s   |
| 2,700         | 99.74%       | employees.views:filtering   |  | 0.002 s   |
| 2,680         | 99.11%       | Render/employees/filtering.html   |  | 0.004 s   |
| 2,680         | 99.08%       | Render/base.html  |  | 0.005 s   |
| 0             | 0.00%        | Block/title   |  | 0.005 s   |
| 0             | 0.00%        | Block/extra_css   |  | 0.005 s   |
| 0             | 0.00%        | Block/page_title  |  | 0.006 s   |
| 2,680         | 99.00%       | Block/content   |  | 0.006 s   |
| 852           | 31.46%       | psycopg2:connect  |  | 0.006 s   |
| 777           | 28.69%       | employees_employee - SELECT   |  | 0.935 s   |
| 975           | 36.00%       | Application code (in Block/content)  |  | 1.712 s   |
| 1.0           | 0.04%        | Block/extra_js  |  | 2.687 s   |
| 2.0           | 0.07%        | 5 fast method calls   |  | 2.703 s   |
| 3.0           | 0.11%        | WSGI/Response   |  | 2.705 s   |

???



# NON-INSTRUMENTED CODE



# HYBRID PROFILING

PROs

CONs

FAST

SEMI-MANUAL

FUNCTION ARGS

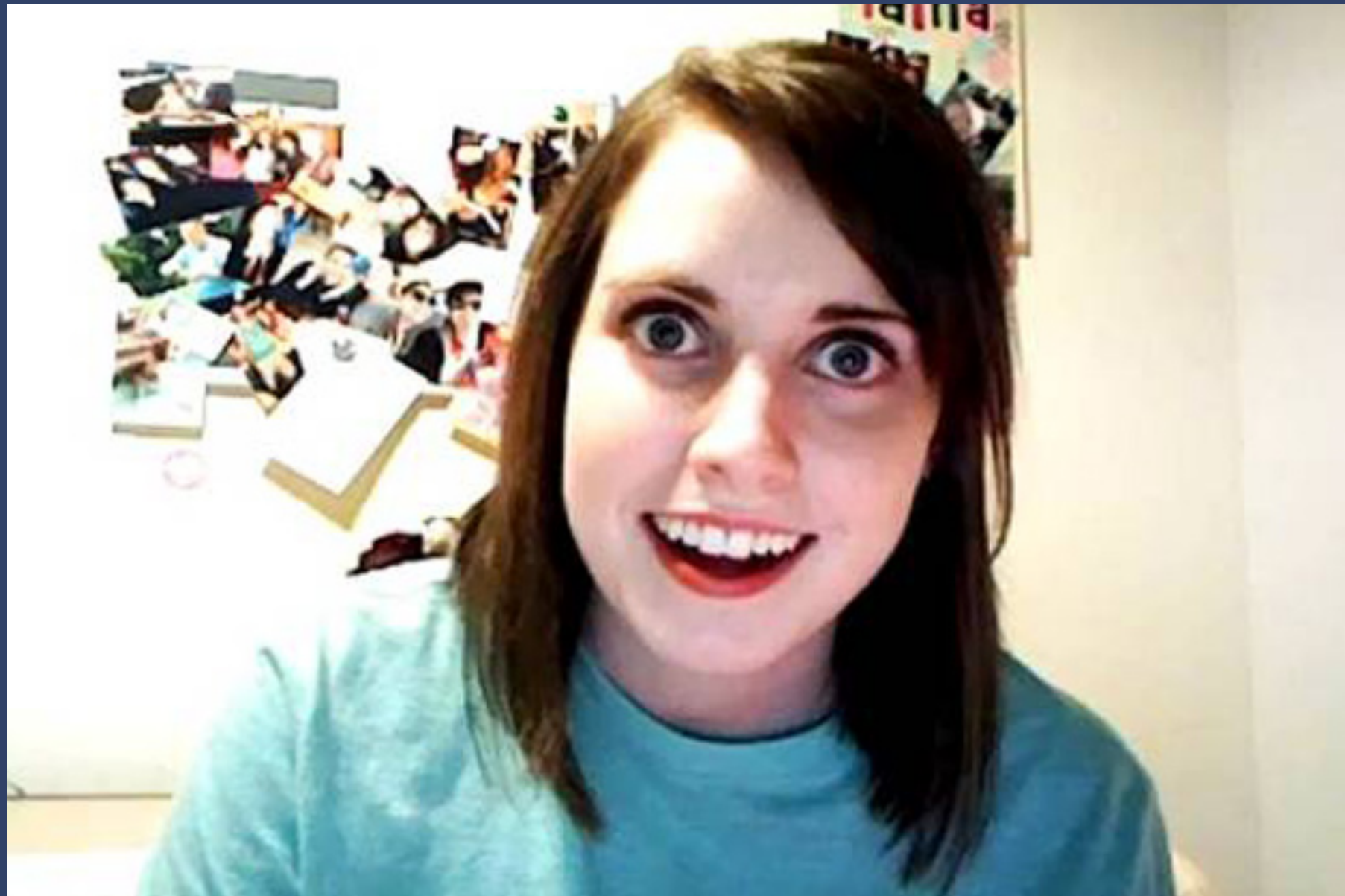
LIMITED  
INSTRUMENTATION



# STATISTICAL PROFILER



@amjithr



@amjithr



WAT?



@amjithr

OVERLY ATTACHED



@amjithr

Interrupt

Inquire

Collate

OVERLY ATTACHED



@amjithr

Interrupt

Inquire

Collate



OVERLY ATTACHED



@amjithr



Interrupt



Inquire



Collate

OVERLY ATTACHED



Interrupt



Inquire



Collate



OVERLY ATTACHED



Interrupt



Inquire



Collate



# STATISTICAL PROFILER



# INTERRUPT

UNIX SIGNALS  
THREADS



# INQUIRE

Stack frame of every thread



# INQUIRE

```
>>> import sys, traceback
```

```
# Get stack frames of every thread
```

```
>>> frames = sys._current_frames()
```

```
# Extract useful info from frame
```

```
# (file, line_no, function, line)
```

```
>>> traceback.extract_stack(frame)
```

```
[('code.py', 88, 'runsource', 'return'),  
 ('code.py', 110, 'runcode', 'print')]
```



# EXISTING TOOLS



# StatProf

UNIX SIGNALS  
CLI



<http://pypi.python.org/pypi/statprof>

@amjithr

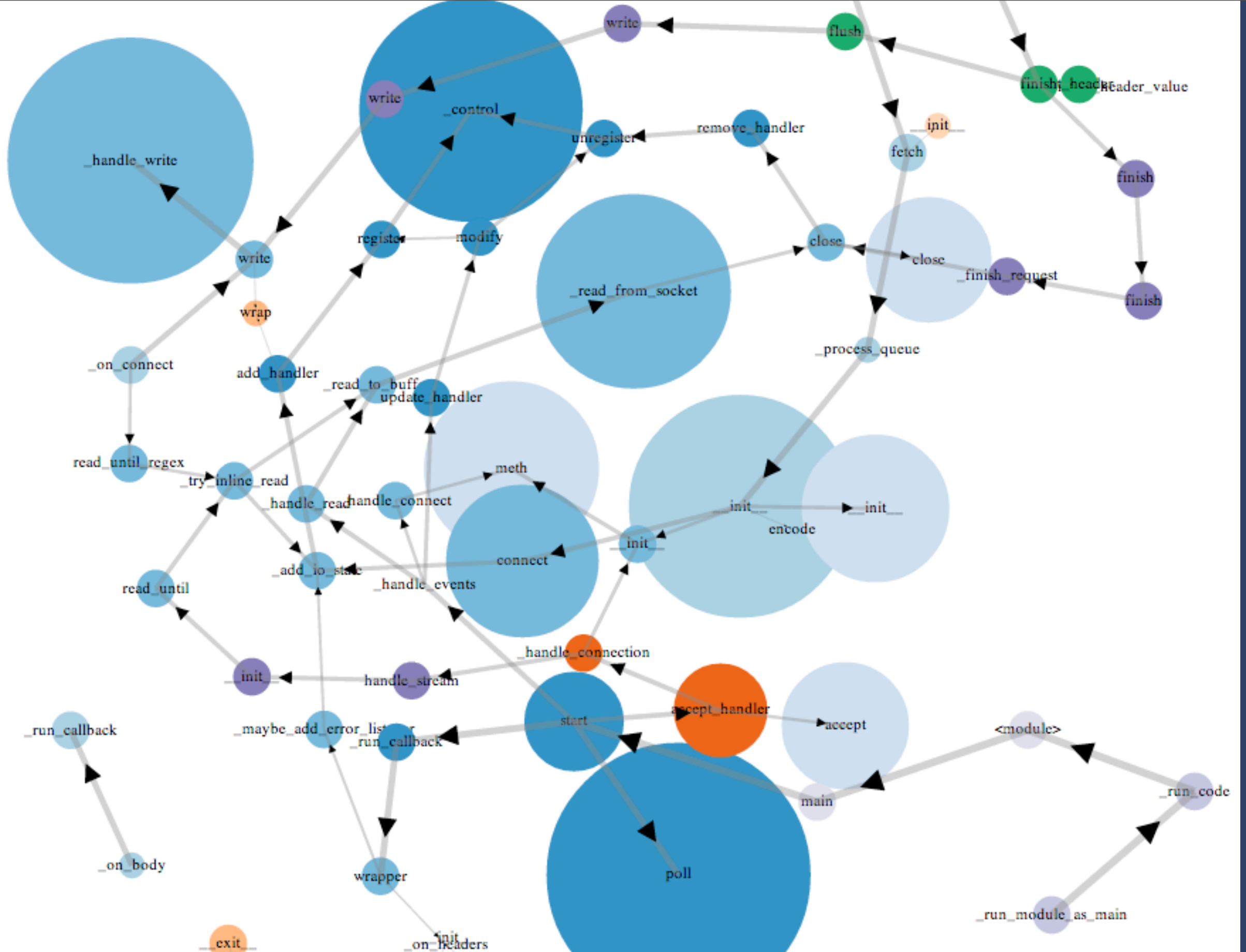


# PLOP

UNIX SIGNALS

D3 Call Graph





# NewRelic

## Threads GUI

## Tree settings

### Show

Web request

Background

Other

### Orientation

Top down view

Bottom up view

"Top-down" shows call stacks in the traditional parent→child order

### Granularity

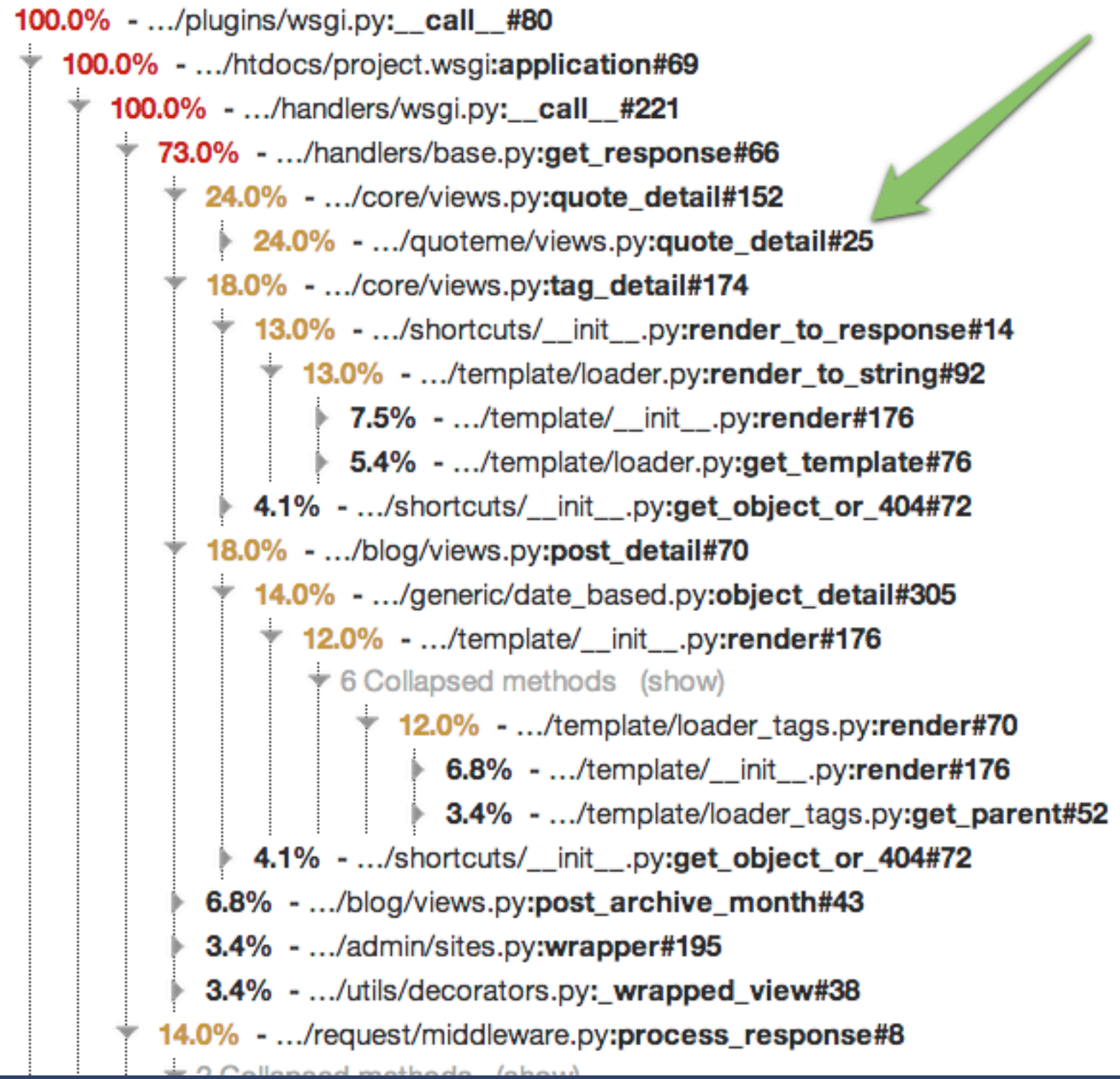
Filter outliers

### Line numbers

Show line numbers

Refresh tree

Expand most called Expand all Collapse all



# FINALE

## DETERMINISTIC + STATISTICAL



@amjithr

# X-Ray Sessions

~~(SECRET BETA)~~

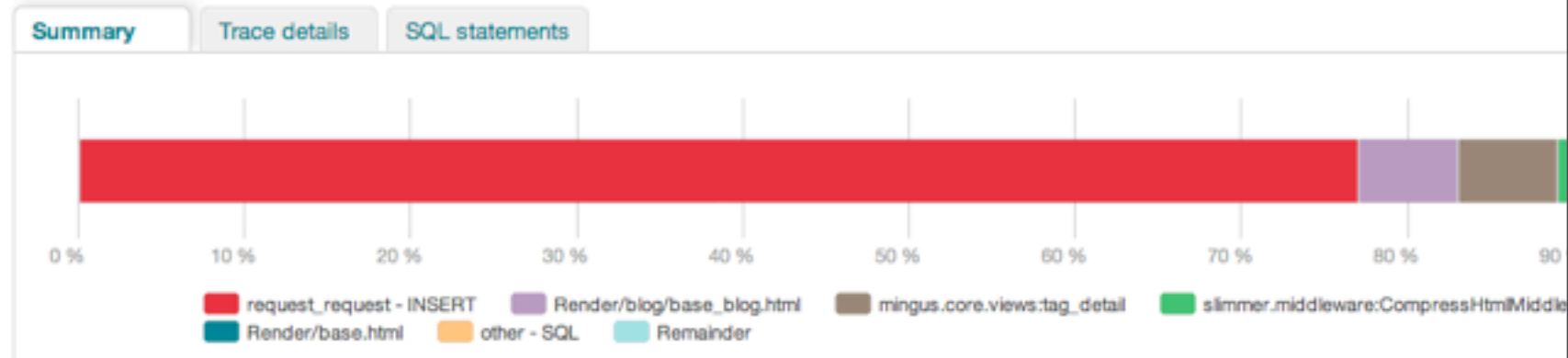


@amjithr

| Sample performance details | App Server |
|----------------------------|------------|
| 09:31 — about 12 hours ago | 108.0 ms   |
| 09:32 — about 12 hours ago | 83.0 ms    |
| 09:32 — about 12 hours ago | 71.0 ms    |
| 09:29 — about 12 hours ago | 68.0 ms    |
| 09:33 — about 12 hours ago | 68.0 ms    |
| 09:30 — about 12 hours ago | 49.0 ms    |
| 09:30 — about 12 hours ago | 47.0 ms    |
| 09:30 — about 12 hours ago | 43.0 ms    |
| 09:29 — about 12 hours ago | 42.0 ms    |
| 09:35 — about 12 hours ago | 40.0 ms    |
| 09:34 — about 12 hours ago | 38.0 ms    |
| 09:34 — about 12 hours ago | 38.0 ms    |
| 09:29 — about 12 hours ago | 37.0 ms    |
| 09:37 — about 12 hours ago | 37.0 ms    |
| 09:31 — about 12 hours ago | 36.0 ms    |
| 09:32 — about 12 hours ago | 35.0 ms    |
| 09:34 — about 12 hours ago | 35.0 ms    |
| 09:31 — about 12 hours ago | 34.0 ms    |
| 09:30 — about 12 hours ago | 33.0 ms    |
| 09:32 — about 12 hours ago | 33.0 ms    |
| 09:37 — about 12 hours ago | 33.0 ms    |

### App server trace on Grumpys-MacBook-Pro-15.local(Mingus-1)

09:31:31 108 ms  
Trace time Resp. time



| Slowest components   | Count |
|--|-------|
| request_request - INSERT                                   | 1     |
| Render/blog/base_blog.html                                 | 1     |
| mingus.core.views.tag_detail                               | 1     |
| slimmer.middleware.CompressHtmlMiddleware.process_response | 1     |
| Render/base.html   | 1     |
| other - SQL  | 1     |
| Remainder  | 1     |
| <b>Total</b>   |       |

Request parameters ?  
URL /tags/django/

# DETERMINISTIC





Overview  
91 Traces

Thread Profile  
Sample Count 6563

Profile collected Today, 9:28  
Duration 1 minute  
Sample count 6563  
Process Grumpys-MacBook-Pro-15.local(Mingus-1)  
Collected by

### Tree settings

#### Show

- Web request
- Background
- Other

#### Orientation

- Top down view
- Bottom up view

"Top-down" shows call stacks in the traditional parent→child order

#### Granularity

- Filter outliers

#### Line numbers

- Show line numbers

Refresh tree



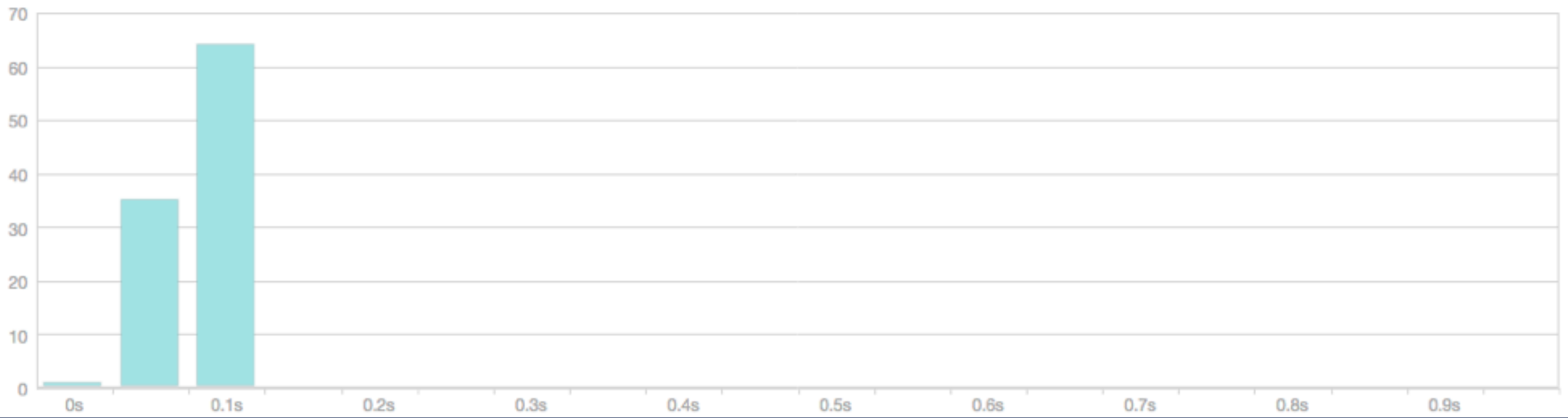
# STATISTICAL



@amjithr

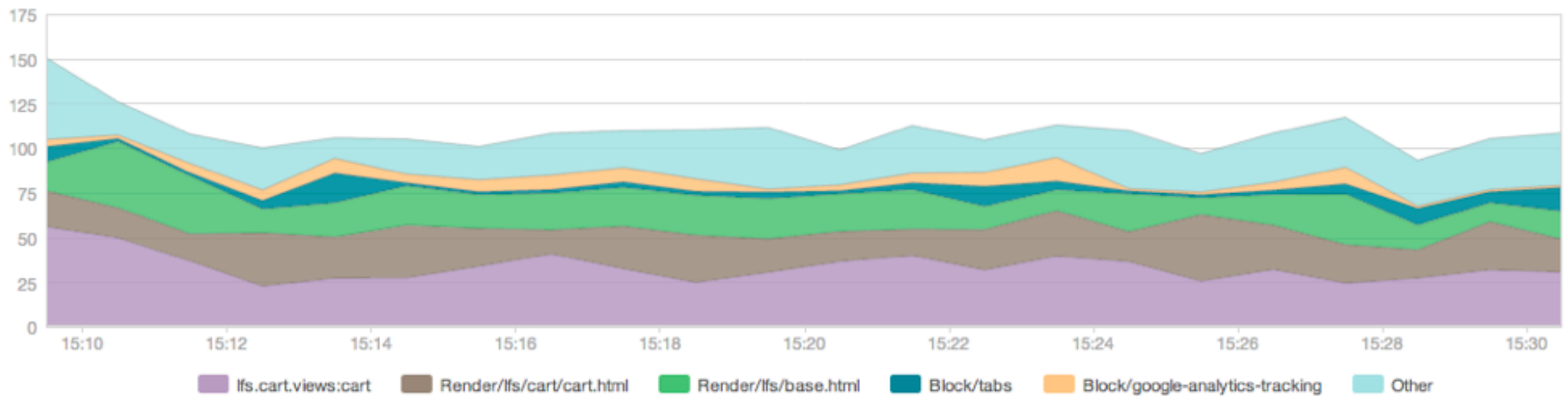


Count of traces per response time



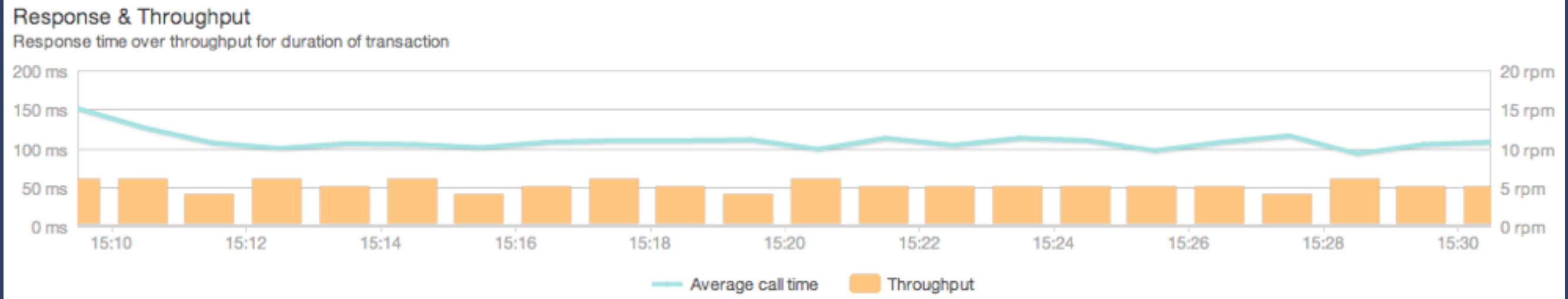
App Server Breakdown

Breakdown of app server response time



# HISTOGRAM





# RESPONSE & THROUGHPUT





# PROFILING IN PRODUCTION



@amjithr



# QUESTIONS



# Questions?

 New Relic.

- Promo Code: **pycon13**
- 30 days of Pro.
- Free T-shirt.
- In-depth data.

