

# DEBUGGING RUBY PERFORMANCE

Aman Gupta  
@tmm1

[speakerdeck.com/u/tmm1/p/debugging-ruby-performance](https://speakerdeck.com/u/tmm1/p/debugging-ruby-performance)

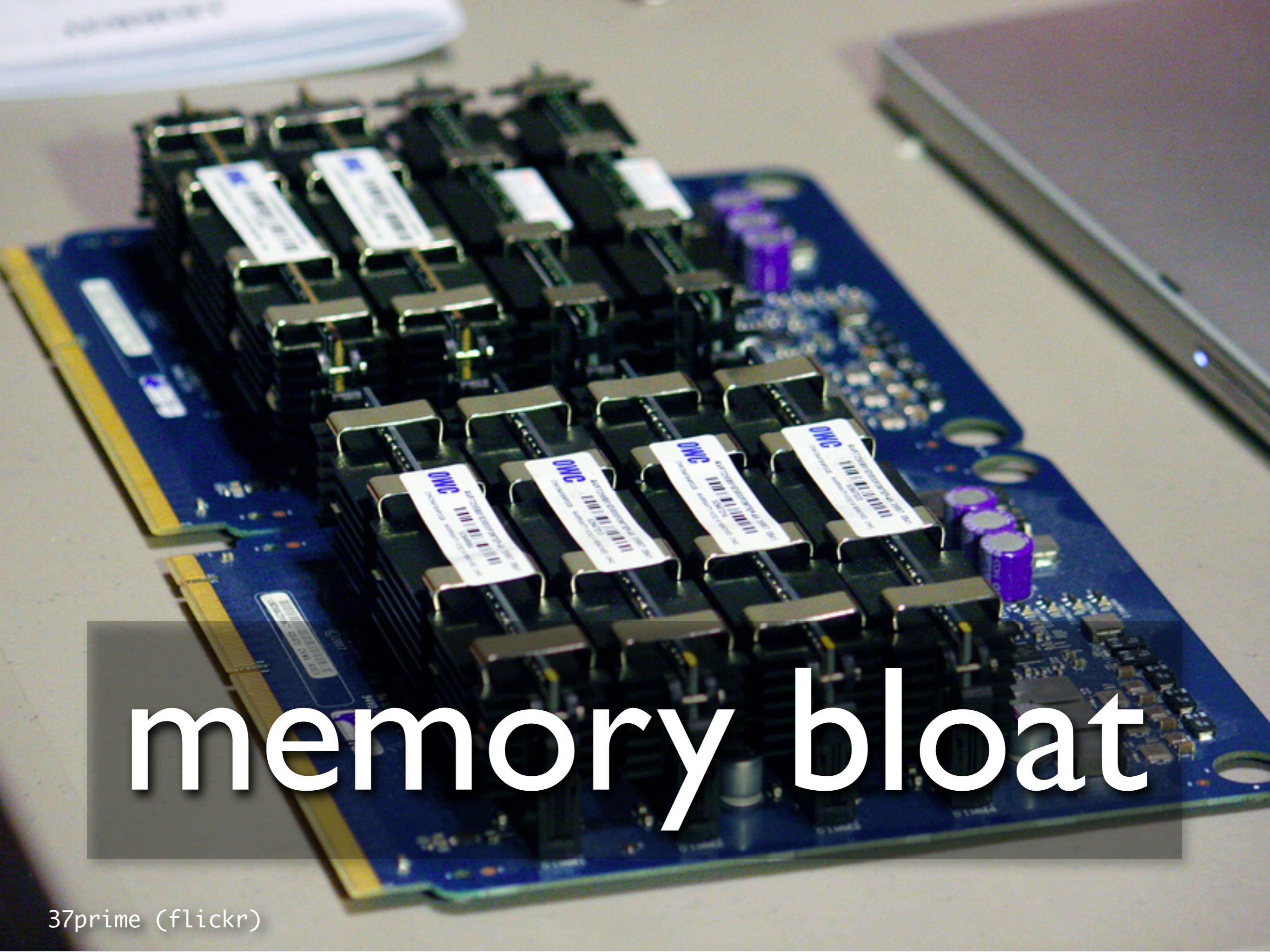
AS RUBYISTS, WE'VE SEEN...



nasty bugs



code



memory bloat



THIS TALK IS ABOUT...

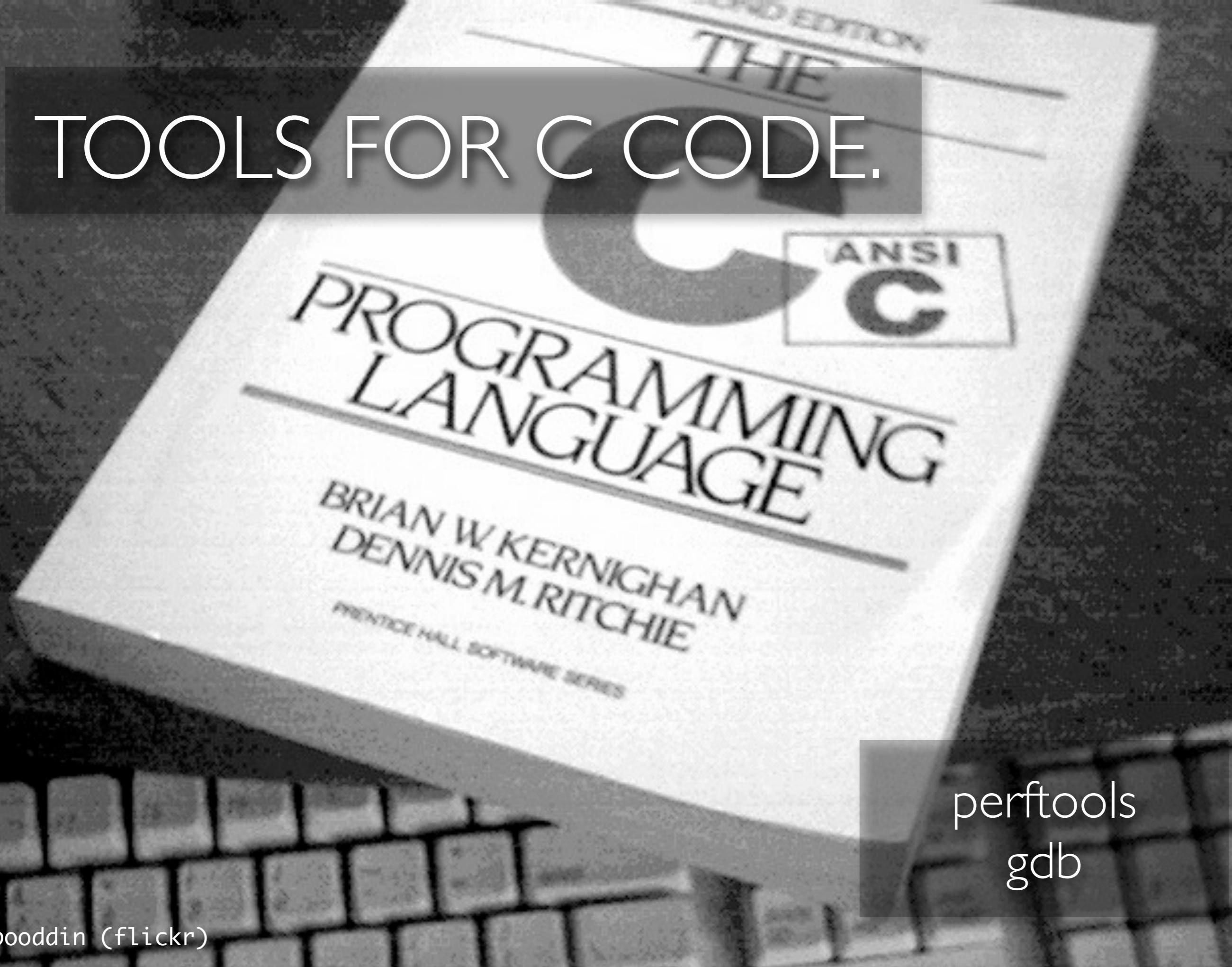
TOOLS TO FIX THESE ISSUES.

# TOOLS FOR LINUX.



lsof  
strace  
ltrace

# TOOLS FOR C CODE.



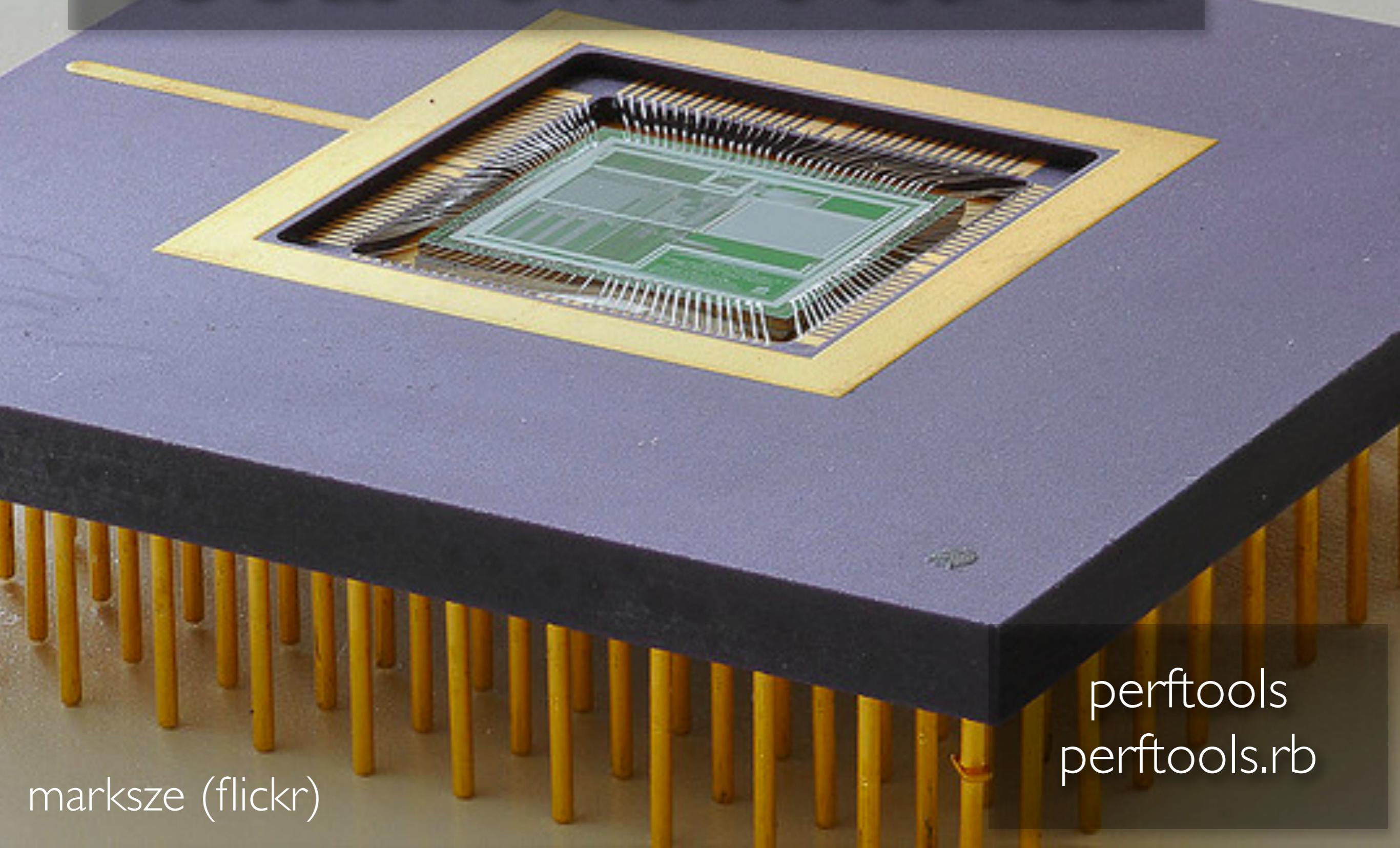
perftools  
gdb

# TOOLS FOR NETWORKS.



tcpdump  
ngrep

# TOOLS FOR CPU USAGE.



marksze (flickr)

perftools  
perftools.rb

# TOOLS FOR MEMORY USAGE.



krocki (flickr)

bleak\_house  
gdb.rb  
memprof



# LSOF

## list open files

```
lsof -nPp <pid>
```

# lsof -nPp <pid>

-n

Inhibits the conversion of network numbers to host names.

-P

Inhibits the conversion of port numbers to names for network files

| FD  | TYPE | NAME  |
|-----|------|---|
| cwd | DIR  | /var/www/myapp                                    |
| txt | REG  | /usr/bin/ruby                                     |
| mem | REG  | /json-1.1.9/ext/json/ext/generator.so             |
| mem | REG  | /json-1.1.9/ext/json/ext/parser.so                |
| mem | REG  | /memcached-0.17.4/lib/rlibmemcached.so            |
| mem | REG  | /mysql-2.8.1/lib/mysql_api.so                     |
| 0u  | CHR  | /dev/null   |
| 1w  | REG  | /usr/local/nginx/logs/error.log                   |
| 2w  | REG  | /usr/local/nginx/logs/error.log                   |
| 3u  | IPv4 | 10.8.85.66:33326->10.8.85.68:3306 (ESTABLISHED)   |
| 10u | IPv4 | 10.8.85.66:33327->10.8.85.68:3306 (ESTABLISHED)   |
| 11u | IPv4 | 127.0.0.1:58273->127.0.0.1:11211 (ESTABLISHED)    |
| 12u | REG  | /tmp/RackMultipart.28957.0                        |
| 33u | IPv4 | 174.36.83.42:37466->69.63.180.21:80 (ESTABLISHED) |

# lsof -nPp <pid>

-n

Inhibits the conversion of network numbers to host names.

-P

Inhibits the conversion of port numbers to names for network files

| FD  | TYPE | NAME  | json |
|-----|------|---|------|
| cwd | DIR  | /var/www/myapp                                    |      |
| txt | REG  | /usr/bin/ruby                                     |      |
| mem | REG  | /json-1.1.9/ext/json/ext/generator.so             |      |
| mem | REG  | /json-1.1.9/ext/json/ext/parser.so                |      |
| mem | REG  | /memcached-0.17.4/lib/rlibmemcached.so            |      |
| mem | REG  | /mysql-2.8.1/lib/mysql_api.so                     |      |
| 0u  | CHR  | /dev/null   |      |
| 1w  | REG  | /usr/local/nginx/logs/error.log                   |      |
| 2w  | REG  | /usr/local/nginx/logs/error.log                   |      |
| 3u  | IPv4 | 10.8.85.66:33326->10.8.85.68:3306 (ESTABLISHED)   |      |
| 10u | IPv4 | 10.8.85.66:33327->10.8.85.68:3306 (ESTABLISHED)   |      |
| 11u | IPv4 | 127.0.0.1:58273->127.0.0.1:11211 (ESTABLISHED)    |      |
| 12u | REG  | /tmp/RackMultipart.28957.0                        |      |
| 33u | IPv4 | 174.36.83.42:37466->69.63.180.21:80 (ESTABLISHED) |      |

# lsof -nPp <pid>

-n

Inhibits the conversion of network numbers to host names.

-P

Inhibits the conversion of port numbers to names for network files

| FD  | TYPE | NAME  |                  |
|-----|------|---|------------------|
| cwd | DIR  | /var/www/myapp                                    | <b>json</b>      |
| txt | REG  | /usr/bin/ruby                                     | <b>memcached</b> |
| mem | REG  | /json-1.1.9/ext/json/ext/generator.so             |                  |
| mem | REG  | /json-1.1.9/ext/json/ext/parser.so                |                  |
| mem | REG  | /memcached-0.17.4/lib/libmemcached.so             |                  |
| mem | REG  | /mysql-2.8.1/lib/mysql_api.so                     |                  |
| 0u  | CHR  | /dev/null   |                  |
| 1w  | REG  | /usr/local/nginx/logs/error.log                   |                  |
| 2w  | REG  | /usr/local/nginx/logs/error.log                   |                  |
| 3u  | IPv4 | 10.8.85.66:33326->10.8.85.68:3306 (ESTABLISHED)   |                  |
| 10u | IPv4 | 10.8.85.66:33327->10.8.85.68:3306 (ESTABLISHED)   |                  |
| 11u | IPv4 | 127.0.0.1:58273->127.0.0.1:11211 (ESTABLISHED)    |                  |
| 12u | REG  | /tmp/RackMultipart.28957.0                        |                  |
| 33u | IPv4 | 174.36.83.42:37466->69.63.180.21:80 (ESTABLISHED) |                  |

# lsof -nPp <pid>

-n

Inhibits the conversion of network numbers to host names.

-P

Inhibits the conversion of port numbers to names for network files

| FD  | TYPE | NAME  |           |
|-----|------|---|-----------|
| cwd | DIR  | /var/www/myapp                                    | json      |
| txt | REG  | /usr/bin/ruby                                     | memcached |
| mem | REG  | /json-1.1.9/ext/json/ext/generator.so             | mysql     |
| mem | REG  | /json-1.1.9/ext/json/ext/parser.so                |           |
| mem | REG  | /memcached-0.17.4/lib/libmemcached.so             |           |
| mem | REG  | /mysql-2.8.1/lib/mysql_api.so                     |           |
| 0u  | CHR  | /dev/null   |           |
| 1w  | REG  | /usr/local/nginx/logs/error.log                   |           |
| 2w  | REG  | /usr/local/nginx/logs/error.log                   |           |
| 3u  | IPv4 | 10.8.85.66:33326->10.8.85.68:3306 (ESTABLISHED)   |           |
| 10u | IPv4 | 10.8.85.66:33327->10.8.85.68:3306 (ESTABLISHED)   |           |
| 11u | IPv4 | 127.0.0.1:58273->127.0.0.1:11211 (ESTABLISHED)    |           |
| 12u | REG  | /tmp/RackMultipart.28957.0                        |           |
| 33u | IPv4 | 174.36.83.42:37466->69.63.180.21:80 (ESTABLISHED) |           |

# lsof -nPp <pid>

-n

Inhibits the conversion of network numbers to host names.

-P

Inhibits the conversion of port numbers to names for network files

| FD  | TYPE | NAME  |           |
|-----|------|---|-----------|
| cwd | DIR  | /var/www/myapp                                    | json      |
| txt | REG  | /usr/bin/ruby                                     | memcached |
| mem | REG  | /json-1.1.9/ext/json/ext/generator.so             | mysql     |
| mem | REG  | /json-1.1.9/ext/json/ext/parser.so                | http      |
| mem | REG  | /memcached-0.17.4/lib/libmemcached.so             |           |
| mem | REG  | /mysql-2.8.1/lib/mysql_api.so                     |           |
| 0u  | CHR  | /dev/null   |           |
| 1w  | REG  | /usr/local/nginx/logs/error.log                   |           |
| 2w  | REG  | /usr/local/nginx/logs/error.log                   |           |
| 3u  | IPv4 | 10.8.85.66:33326->10.8.85.68:3306 (ESTABLISHED)   |           |
| 10u | IPv4 | 10.8.85.66:33327->10.8.85.68:3306 (ESTABLISHED)   |           |
| 11u | IPv4 | 127.0.0.1:58273->127.0.0.1:11211 (ESTABLISHED)    |           |
| 12u | REG  | /tmp/RackMultipart.28957.0                        |           |
| 33u | IPv4 | 174.36.83.42:37466->69.63.180.21:80 (ESTABLISHED) |           |

# lsof -nPp <pid>

-n

Inhibits the conversion of network numbers to host names.

-P

Inhibits the conversion of port numbers to names for network files

| FD  | TYPE | NAME  |           |
|-----|------|---|-----------|
| cwd | DIR  | /var/www/myapp                                    | json      |
| txt | REG  | /usr/bin/ruby                                     | memcached |
| mem | REG  | /json-1.1.9/ext/json/ext/generator.so             | mysql     |
| mem | REG  | /json-1.1.9/ext/json/ext/parser.so                | http      |
| mem | REG  | /memcached-0.17.4/lib/libmemcached.so             |           |
| mem | REG  | /mysql-2.8.1/lib/mysql_api.so                     |           |
| 0u  | CHR  | /dev/null   |           |
| 1w  | REG  | /usr/local/nginx/logs/error.log                   |           |
| 2w  | REG  | /usr/local/nginx/logs/error.log                   |           |
| 3u  | IPv4 | 10.8.85.66:33326->10.8.85.68:3306 (ESTABLISHED)   |           |
| 10u | IPv4 | 10.8.85.66:33327->10.8.85.68:3306 (ESTABLISHED)   |           |
| 11u | IPv4 | 127.0.0.1:58273->127.0.0.1:11211 (ESTABLISHED)    |           |
| 12u | REG  | /tmp/RackMultipart.28957.0                        |           |
| 33u | IPv4 | 174.36.83.42:37466->69.63.180.21:80 (ESTABLISHED) |           |

# TCPDUMP

dump traffic on a network

```
tcpdump -i eth0 -s 0 -nqA  
      tcp dst port 3306
```

`tcpdump -i <eth> -s <len> -nqA <expr>`

`tcpdump -i <eth> -w <file> <expr>`

`-i <eth>`

Network interface.

`-s <len>`

Snarf len bytes of data from each packet.

`-n`

Don't convert addresses (host addresses, port numbers) to names.

`-q`

Quiet output. Print less protocol information.

`-A`

Print each packet (minus its link level header) in ASCII.

`-w <file>`

Write the raw packets to file rather than printing them out.

`<expr>`

libpcap expression, for example:

`tcp src port 80`

`tcp dst port 3306`

tcp dst port 80

19:52:20.216294 IP 24.203.197.27.40105 >

**174.37.48.236.80**: tcp 438

E...\*.@.l.%&.....%0....P0x..%s.oP.....

**GET /poll\_images/cld99erh0/logo.png HTTP/1.1**

Accept: \*/\*

Referer: [http://apps.facebook.com/realpolls/?fb\\_q=1](http://apps.facebook.com/realpolls/?fb_q=1)

tcp dst port 3306

19:51:06.501632 IP 10.8.85.66.50443 >

10.8.85.68.3306: tcp 98

E... "K@. @. Yy

.UB

.UD.....z....L.....

GZ.y3b..[.....W....

SELECT \* FROM `votes` WHERE (`poll\_id` =  
72621) LIMIT 1

# tcpdump -w <file>

## ▽ Hypertext Transfer Protocol

```
▷ GET / HTTP/1.1\r\n
  Host: 192.168.0.2\r\n
  User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.0; en-US; rv:1.5
  Accept: text/xml,application/xml,application/xhtml+xml,text/html;q
  Accept-Language: en-us,en;q=0.5\r\n
  Accept-Encoding: gzip,deflate\r\n
  Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7\r\n
  Keep-Alive: 300\r\n
```



# STRACE

## trace system calls and signals

```
strace -cp <pid>
strace -ttTp <pid> -o <file>
```

# strace -cp <pid>

-c

Count time, calls, and errors for each system call and report a summary on program exit.

-p pid

Attach to the process with the process ID pid and begin tracing.

| % time | seconds  | usecs/call | calls | errors | syscall    |
|--------|----------|------------|-------|--------|------------|
| 50.39  | 0.000064 | 0          | 1197  | 592    | read       |
| 34.65  | 0.000044 | 0          | 609   |        | writev     |
| 14.96  | 0.000019 | 0          | 1226  |        | epoll_ctl  |
| 0.00   | 0.000000 | 0          | 4     |        | close      |
| 0.00   | 0.000000 | 0          | 1     |        | select     |
| 0.00   | 0.000000 | 0          | 4     |        | socket     |
| 0.00   | 0.000000 | 0          | 4     | 4      | connect    |
| 0.00   | 0.000000 | 0          | 1057  |        | epoll_wait |
| -----  |          |            |       |        |            |
| 100.00 | 0.000127 |            | 4134  | 596    | total      |

# strace -ttTp <pid> -o <file>

-t

Prefix each line of the trace with the time of day.

-tt

If given twice, the time printed will include the microseconds.

-T

Show the time spent in system calls.

-o filename

Write the trace output to the file filename rather than to stderr.

```
epoll_wait(9, {{EPOLLIN, {u32=68841296, u64=68841296}}}, 4096, 50) = 1 <0.033109>
accept(10, {sin_port=38313, sin_addr="127.0.0.1"}, [1226]) = 22 <0.000014>
fcntl(22, F_GETFL) = 0x2 (flags O_RDWR) <0.000007>
fcntl(22, F_SETFL, O_RDWR|O_NONBLOCK) = 0 <0.000008>
setsockopt(22, SOL_TCP, TCP_NODELAY, [1], 4) = 0 <0.000008>
accept(10, 0x7fff5d9c07d0, [1226]) = -1 EAGAIN <0.000014>
epoll_ctl(9, EPOLL_CTL_ADD, 22, {EPOLLIN, {u32=108750368, u64=108750368}}) = 0 <0.000009>
epoll_wait(9, {{EPOLLIN, {u32=108750368, u64=108750368}}}, 4096, 50) = 1 <0.000007>
read(22, "GET / HTTP/1.1\r\n...", 16384) = 772 <0.000012>
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0 <0.000007>
poll([{fd=5, events=POLLIN|POLLPRI}], 1, 0) = 0 (Timeout) <0.000008>
write(5, "1\0\0\0\0\0-\0\0\0\3SELECT * FROM `table`\r\n...", 56) = 56 <0.000023>
read(5, "\25\1\0\1,\2\0x\234m\r\n...", 16384) = 284 <1.300897>
```

http client connection

read 772 bytes

read(22, "GET / HTTP/1.1\r...\n", 16384) = 772 <0.0012>

incoming http request

took 0.0012s

mysql connection

write sql query to db

```
write(5, "SELECT * FROM `table`...", 56) = 56 <0.0023>
read(5, "\25\1\0\1,\2\0x\234m...", 16384) = 284 <1.30>
```

↑  
read query  
response

↑  
slow query

# stracing ruby: SIGVTALRM

```
--- SIGVTALRM (Virtual timer expired) @ 0 (0) ---
rt_sigreturn(0x1a)      = 2207807 <0.000009>
--- SIGVTALRM (Virtual timer expired) @ 0 (0) ---
rt_sigreturn(0x1a)      = 0 <0.000009>
--- SIGVTALRM (Virtual timer expired) @ 0 (0) ---
rt_sigreturn(0x1a)      = 140734552062624 <0.000009>
--- SIGVTALRM (Virtual timer expired) @ 0 (0) ---
rt_sigreturn(0x1a)      = 140734552066688 <0.000009>
--- SIGVTALRM (Virtual timer expired) @ 0 (0) ---
rt_sigreturn(0x1a)      = 11333952 <0.000008>
--- SIGVTALRM (Virtual timer expired) @ 0 (0) ---
rt_sigreturn(0x1a)      = 0 <0.000009>
--- SIGVTALRM (Virtual timer expired) @ 0 (0) ---
rt_sigreturn(0x1a)      = 1 <0.000010>
--- SIGVTALRM (Virtual timer expired) @ 0 (0) ---
```

- ruby 1.8 uses signals to schedule its green threads
- process receives a SIGVTALRM signal every 10ms

# stracing ruby: sigprocmask

| % time | seconds  | usecs/call | calls   | errors | syscall        |
|--------|----------|------------|---------|--------|----------------|
| 100.00 | 0.326334 | 0          | 3568567 |        | rt_sigprocmask |
| 0.00   | 0.000000 | 0          | 9       |        | read           |
| 0.00   | 0.000000 | 0          | 10      |        | open           |
| 0.00   | 0.000000 | 0          | 10      |        | close          |
| 0.00   | 0.000000 | 0          | 9       |        | fstat          |
| 0.00   | 0.000000 | 0          | 25      |        | mmap           |
| 100.00 | 0.326334 |            | 3568685 | 0      | total          |

- debian/redhat compile ruby with --enable-pthread
- uses a native thread timer for SIGVTALRM
- causes excessive calls to sigprocmask: 30% slowdown!

# stracing ruby: tests

| % time | seconds   | calls | syscall |
|--------|-----------|-------|---------|
| 95.94  | 76.685658 | 1086  | clone   |



95s in the github test suite in clone()  
`git cmd` calls fork() which calls clone()

posix-spawn gem sped up tests by 2x!

# LTRACE

## trace library calls

```
ltrace -cp <pid>  
ltrace -ttTp <pid> -o <file>
```

# ltrace -c ruby threaded\_em.rb

| % time | seconds   | usecs/call | calls | function      |
|--------|-----------|------------|-------|---------------|
| 48.65  | 11.741295 | 617        | 19009 | memcpy        |
| 30.16  | 7.279634  | 831        | 8751  | longjmp       |
| 9.78   | 2.359889  | 135        | 17357 | _setjmp       |
| 8.91   | 2.150565  | 285        | 7540  | malloc        |
| 1.10   | 0.265946  | 20         | 13021 | memset        |
| 0.81   | 0.195272  | 19         | 10105 | __ctype_b_loc |
| 0.35   | 0.084575  | 19         | 4361  | strcmp        |
| 0.19   | 0.046163  | 19         | 2377  | strlen        |
| 0.03   | 0.006272  | 23         | 265   | realloc       |
| 100.00 | 24.134999 |            | 82999 | total         |



memcpy -> memcpy

## ltrace -c ruby threaded\_em.rb

| % time | seconds   | usecs/call | calls | function      |
|--------|-----------|------------|-------|---------------|
| 48.65  | 11.741295 | 617        | 19009 | memcpy        |
| 30.16  | 7.279634  | 831        | 8751  | longjmp       |
| 9.78   | 2.359889  | 135        | 17357 | _setjmp       |
| 8.91   | 2.150565  | 285        | 7540  | malloc        |
| 1.10   | 0.265946  | 20         | 13021 | memset        |
| 0.81   | 0.195272  | 19         | 10105 | __ctype_b_loc |
| 0.35   | 0.084575  | 19         | 4361  | strcmp        |
| 0.19   | 0.046163  | 19         | 2377  | strlen        |
| 0.03   | 0.006272  | 23         | 265   | realloc       |
| 100.00 | 24.134999 |            | 82999 | total         |

## ltrace -ttT -e memcpy ruby threaded\_em.rb

```
01:24:48.769408 --- SIGVTALRM (Virtual timer expired) ---
01:24:48.769616 memcpy(0x1216000, "", 1086328) = 0x1216000 <0.000578>
01:24:48.770555 memcpy(0x6e32670, "\240\343v", 1086328) = 0x6e32670 <0.000418>

01:24:49.899414 --- SIGVTALRM (Virtual timer expired) ---
01:24:49.899490 memcpy(0x1320000, "", 1082584) = 0x1320000 <0.000628>
01:24:49.900474 memcpy(0x6e32670, "", 1086328) = 0x6e32670 <0.000479>
```

# RBTRACE

## trace ruby method calls

```
require 'rbtrace'  
rbtrace -p <pid> -m <method>
```

```
rbtrace -p <PID> -m  
"Unicorn::App#call(env['PATH_INFO'])"  
"Smoke::Git#(args)"
```

```
Unicorn::App#call(env['PATH_INFO']="/cakephp/cakephp/downloads")  
Smoke::Git#cat_ref(args=["2.0.0-dev"]) <0.046590>  
Smoke::Git#cat_ref(args=["1.3-dev"]) <0.053170>  
Smoke::Git#cat_ref(args=["1.3.7"]) <0.046643>  
Smoke::Git#cat_ref(args=["1.3.6"]) <0.045075>  
Smoke::Git#cat_ref(args=["1.3.5"]) <0.043925>  
Smoke::Git#cat_ref(args=["1.3.4"]) <0.054496>  
Smoke::Git#cat_ref(args=["1.3.3"]) <0.045860>
```



downloads page makes N+1 git calls to  
resolve tags

```
$ rbtrace -p 20052 --slow=50
*** attached to process 20052
Array#include? <0.075737>
Array#include? <0.077302>
Array#include? <0.076444>
Array#include? <0.077122>
Array#include? <0.074956>
```

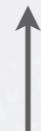
```
$ rbtrace -p 20052 --slow=50
*** attached to process 20052
Array#include? <0.075737>
Array#include? <0.077302>
Array#include? <0.076444>
Array#include? <0.077122>
Array#include? <0.074956>
```

```
$ rbtrace -p 20052 --slow=50 -m 'include?(__source__)'
*** attached to process 20052
Array#include?(__source__="lib/jekyll/site.rb:179") <0.075737>
Array#include?(__source__="lib/jekyll/site.rb:179") <0.077302>
Array#include?(__source__="lib/jekyll/site.rb:179") <0.076444>
```



slow jekyll processing fixed by switching to  
Set#include?

```
Albino#colorize(@target.size=180, @options[:l]="plain") <0.151861>
Albino#colorize(@target.size=1009, @options[:l]="ruby") <0.233931>
Albino#colorize(@target.size=9024, @options[:l]="js") <0.278415>
Albino#colorize(@target.size=243, @options[:l]="plain") <0.155549>
Albino#colorize(@target.size=596, @options[:l]="diff") <0.169390>
Albino#colorize(@target.size=2994, @options[:l]="ruby") <0.305385>
Albino#colorize(@target.size=2372, @options[:l]="js") <0.227131>
Albino#colorize(@target.size=42444, @options[:l]="php") <0.448636>
Albino#colorize(@target.size=1795, @options[:l]="ruby") <0.259931>
Albino#colorize(@target.size=3088, @options[:l]="plain") <0.480678>
```



~500ms to shell out to python *pygmentize*

use pygments.rb for 15x faster syntax highlighting!  
(uses FFI to embed python inside ruby)

```
rbtrace -c <predefined tracer>
```

```
MysqlAdapter#execute(sql="SELECT * FROM `users` WHERE (`users`.`id` = 105775) ") <0.000472>
MysqlAdapter#execute(sql="SELECT * FROM `gists` WHERE (`gists`.`public` = 1 AND `gists`.`id` = '871708' ) LIMIT 1") <0.002153>
MysqlAdapter#execute(sql="SELECT count(*) AS count_all FROM `gist_comments` WHERE (`gist_comments`.gist_id = 871708) ") <0.000866>
```

```
Redis::Client#process(commands=[[:zscore, "gwiki:597840:updated", ["home"]]]) <0.000439>
Redis::Client#process(commands=[[:get, "gh.storage.network.1745996"]]) <0.000923>
Redis::Client#process(commands=[[:sismember, "gh:beta:users", "14850"]]) <0.000520>
Redis::Client#process(commands=[[:rpush, "resque:queue:high", "{...}"]]) <0.000428>
```

built-in tracers for  
activerecord, redis, mongo, eventmachine, unicorn,  
testunit

# PERFTOOLS

google's performance tools

```
CPUPROFILE=/tmp/myprof ./myapp  
pprof ./myapp /tmp/myprof
```

```
wget http://google-perf-tools.googlecode.com/files/google-perf-tools-1.6.tar.gz ← download
```

```
tar zxvf google-perf-tools-1.6.tar.gz  
cd google-perf-tools-1.6
```

```
./configure --prefix=/opt
```

```
make ← compile
```

```
sudo make install
```

```
# for linux
```

```
export LD_PRELOAD=/opt/lib/libprofiler.so ← setup
```

```
# for osx
```

```
export DYLD_INSERT_LIBRARIES=/opt/lib/libprofiler.dylib
```

```
CPUPROFILE=/tmp/ruby.prof ruby -e' ← profile
```

```
5_000_000.times{ "hello world" }
```

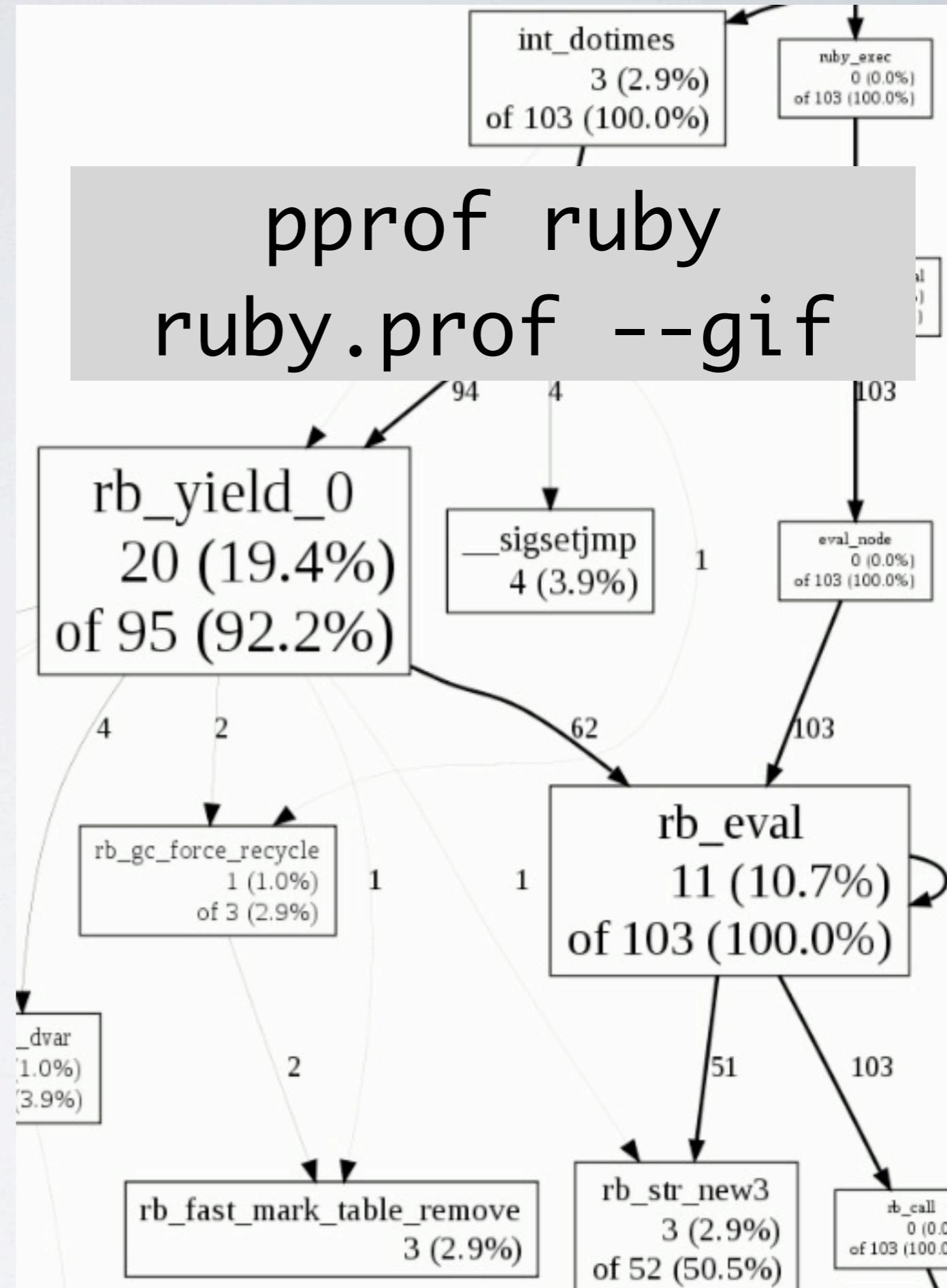
```
,
```

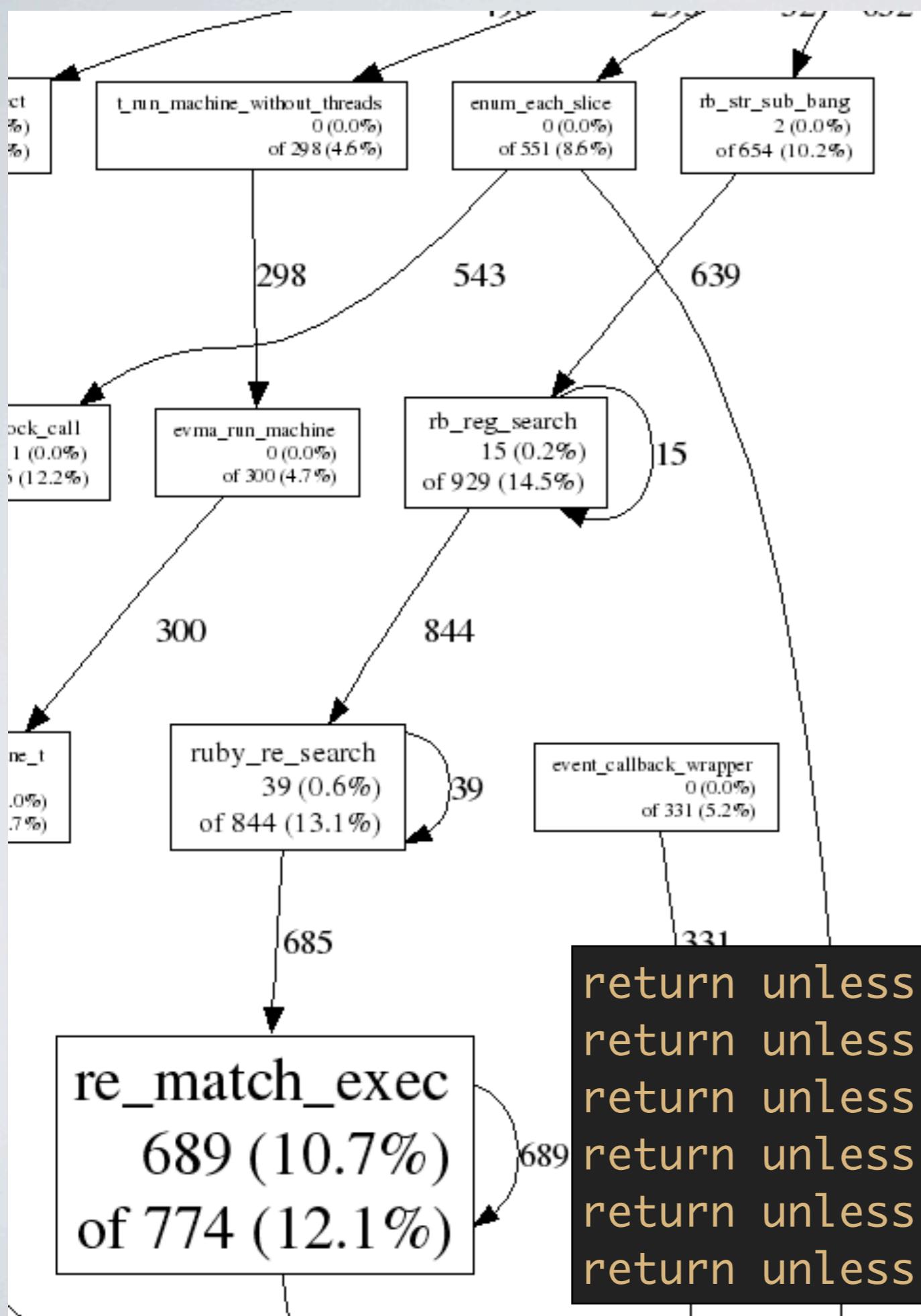
```
pprof `which ruby` --text /tmp/ruby.prof ← report
```

```
pprof ruby  
ruby.prof --text
```

Total: 103 samples

|     |        |             |
|-----|--------|-------------|
| 95  | 92.2%  | rb_yield_0  |
| 103 | 100.0% | rb_eval     |
| 12  | 11.7%  | gc_sweep    |
| 52  | 50.5%  | rb_str_new3 |
| 3   | 2.9%   | obj_free    |
| 103 | 100.0% | int_dotimes |
| 12  | 11.7%  | gc_mark     |





# Profiling MRI

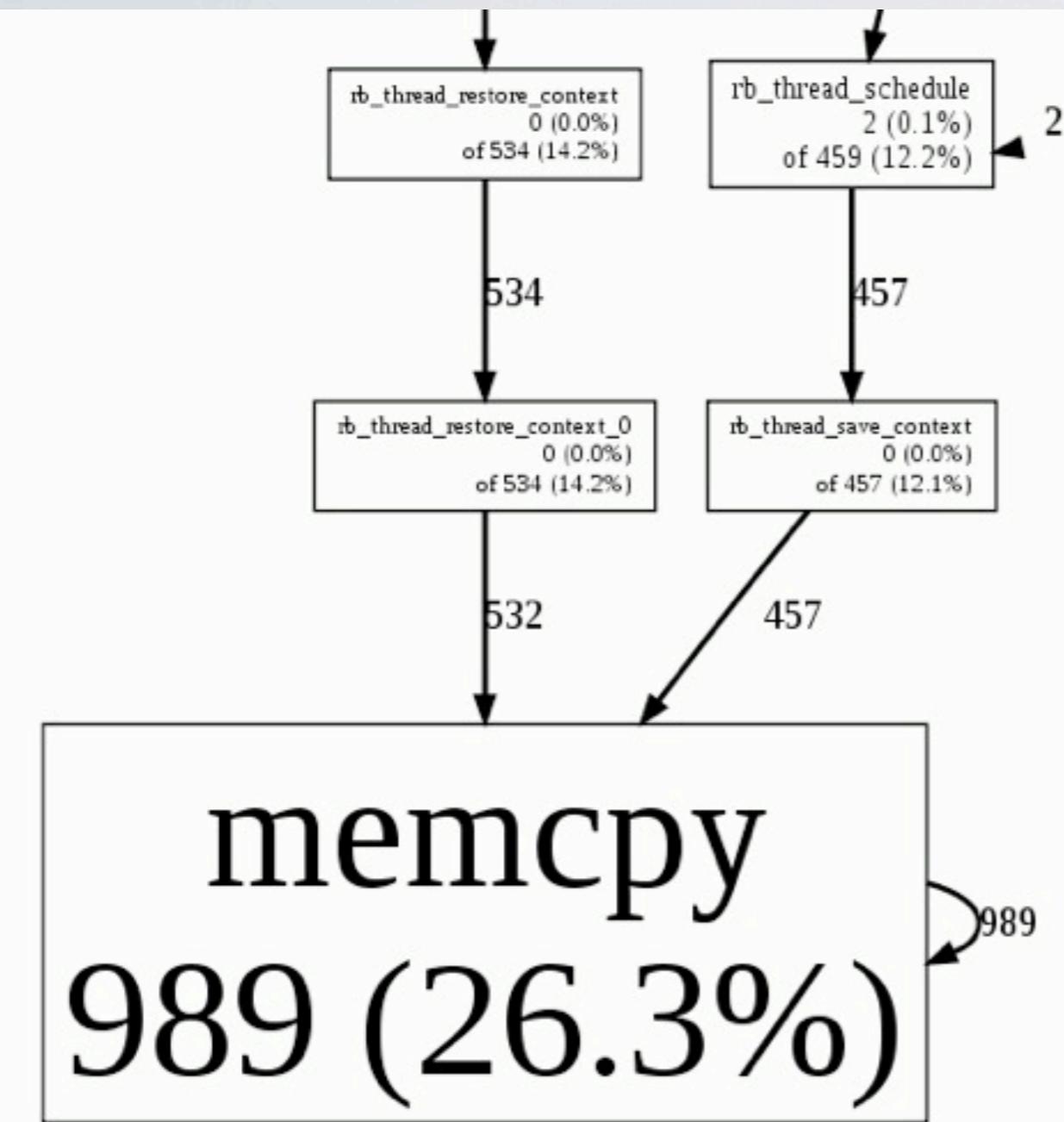
- 10% of production VM time spent in `rb_str_sub_bang`
- `String#sub!`
- called from `Time.parse`

```

return unless str.sub!(/A(\d{1,2})/, '')
return unless str.sub!(/A( \d|\d{1,2})/, '')
return unless str.sub!(/A( \d|\d{1,2})/, '')
return unless str.sub!(/A(\d{1,3})/, '')
return unless str.sub!(/A(\d{1,2})/, '')
return unless str.sub!(/A(\d{1,2})/, '')

```

# Profiling EM + threads



Total: 3763 samples  
2764 73.5% `catch_timer`  
989 26.3% `memcpy`  
3 0.1% `st_lookup`  
2 0.1% `rb_thread_schedule`  
1 0.0% `rb_eval`  
1 0.0% `rb_newobj`  
1 0.0% `rb_gc_force_recycle`

- known issue: EM+threads = slow
- memcpy??
- thread context switches copy the stack w/ memcpy
- EM allocates huge buffer on the stack
- solution: move buffer to the heap

# PERFTOOLS.RB

perftools for ruby code

pprof.rb /tmp/myrbprof

[github.com/tmm1/perftools.rb](https://github.com/tmm1/perftools.rb)

```
gem install perftools.rb
```

```
RUBYOPT="-r`gem which perftools | tail -1`"  
CPUPROFILE=/tmp/myrbprof  
ruby myapp.rb
```

```
pprof.rb /tmp/myrbprof --text  
pprof.rb /tmp/myrbprof --gif > /tmp/myrbprof.gif
```

```

require 'sinatra'

get '/sleep' do
  sleep 0.25
  'done'
end

get '/compute' do
  proc{ |n|
    a,b=0,1
    n.times{ a,b = b,a+b }
    b
  }.call(10_000)
  'done'
end

```

```

$ ab -c 1 -n 50 http://127.0.0.1:4567/compute
$ ab -c 1 -n 50 http://127.0.0.1:4567/sleep

```

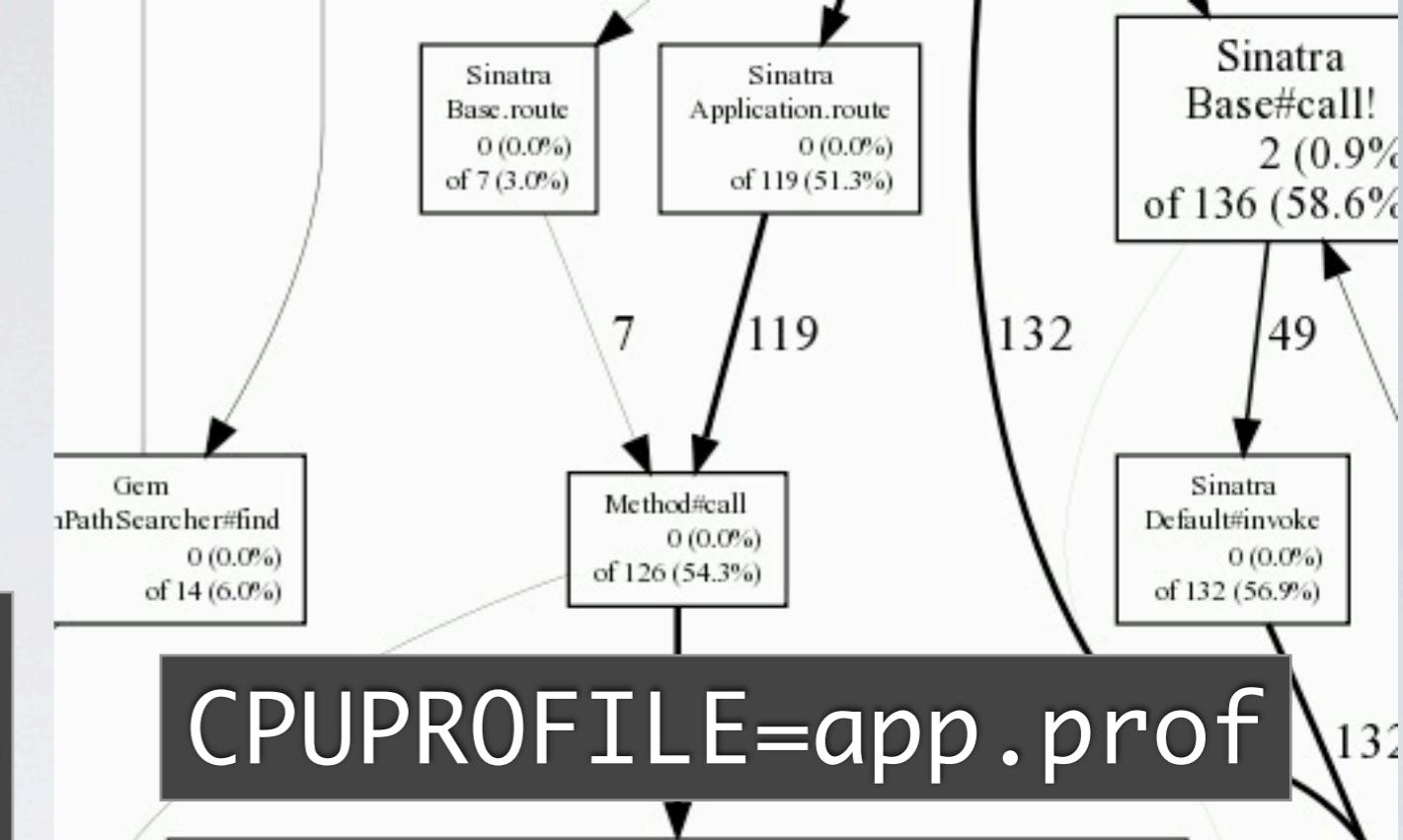
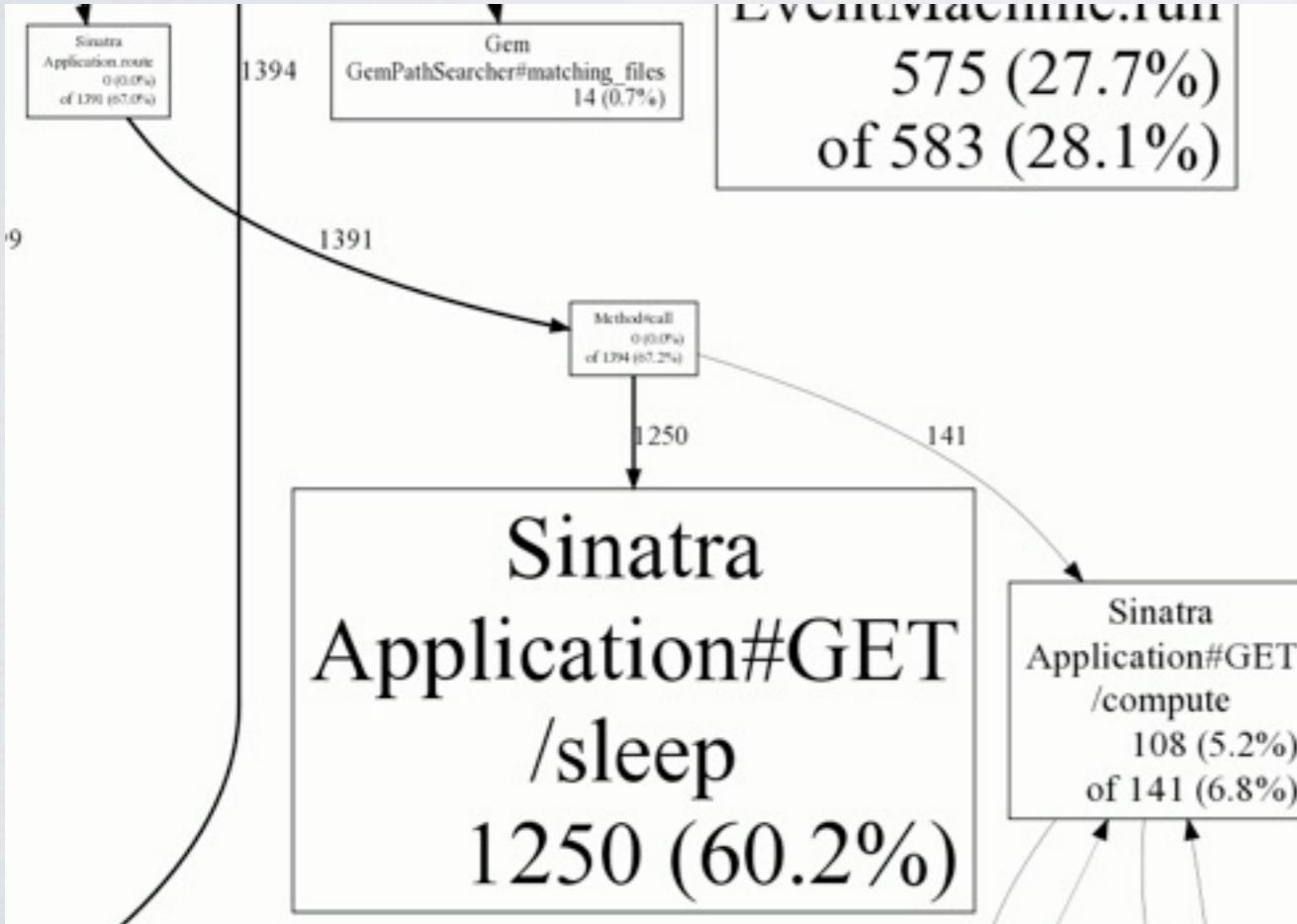
- Sampling profiler:
  - 232 samples total
  - 83 samples were in /compute
  - 118 samples had /compute on the stack but were in another function
  - /compute accounts for 50% of process, but only 35% of time was in /compute itself

== Sinatra has ended his set (crowd applauds)  
 PROFILE: interrupts/evictions/bytes = 232/0/2152

Total: 232 samples

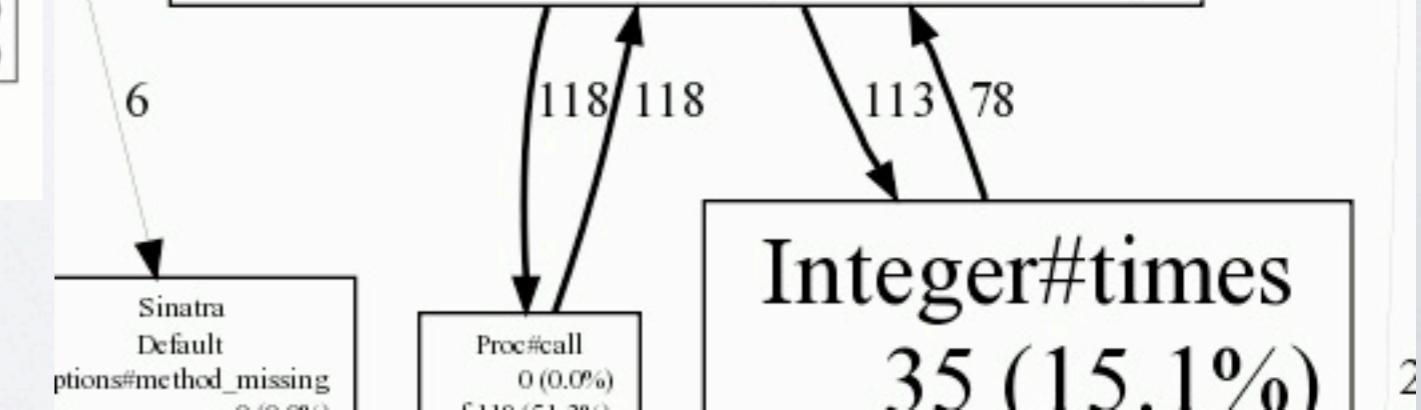
|    |       |       |     |       |                                   |
|----|-------|-------|-----|-------|-----------------------------------|
| 83 | 35.8% | 35.8% | 118 | 50.9% | Sinatra::Application#GET /compute |
| 56 | 24.1% | 59.9% | 56  | 24.1% | garbage_collector                 |
| 35 | 15.1% | 75.0% | 113 | 48.7% | Integer#times                     |

CPUPROFILE\_REALTIME=1  
CPUPROFILE=app-rt.prof

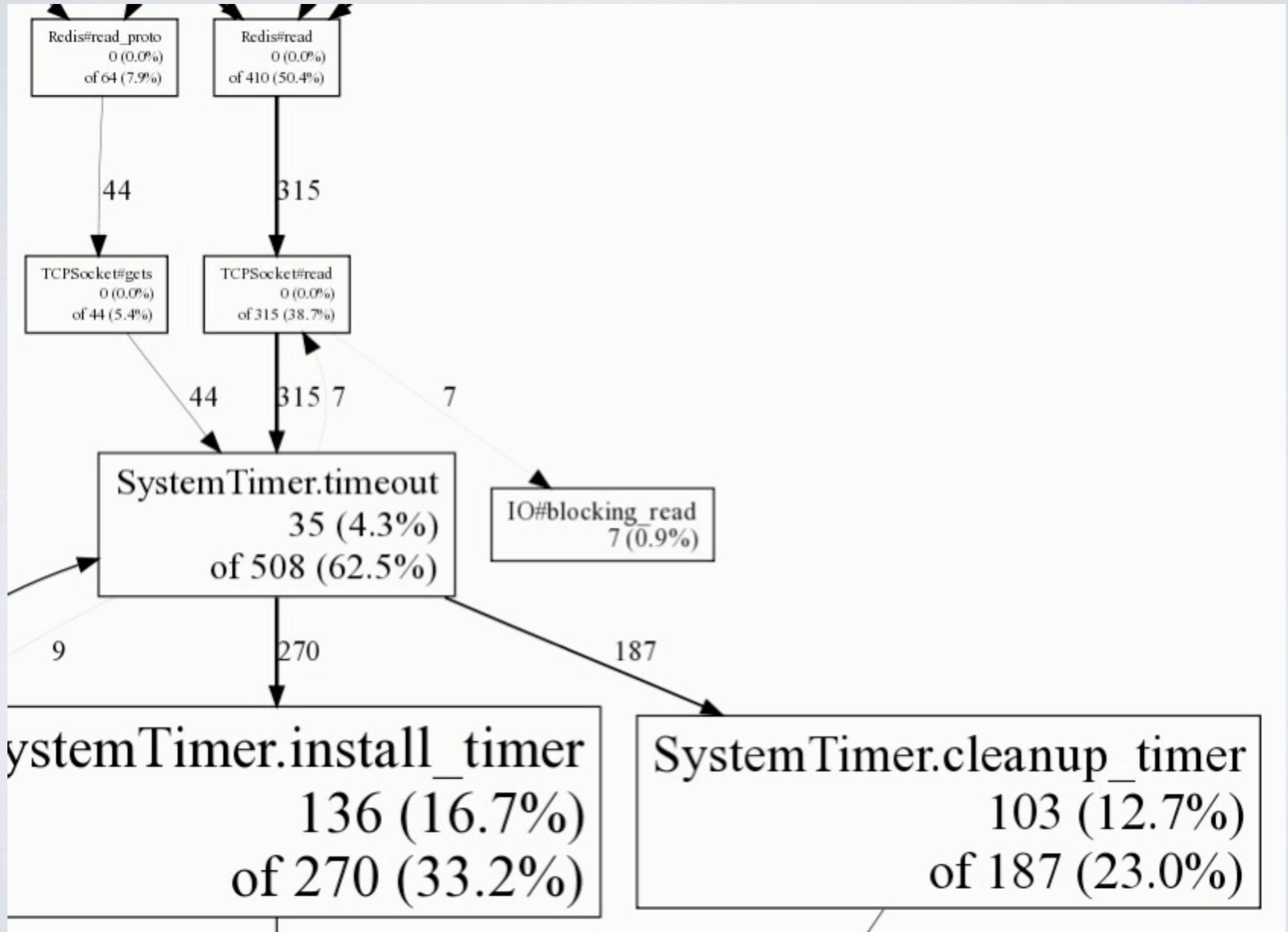


CPUPROFILE=app.prof

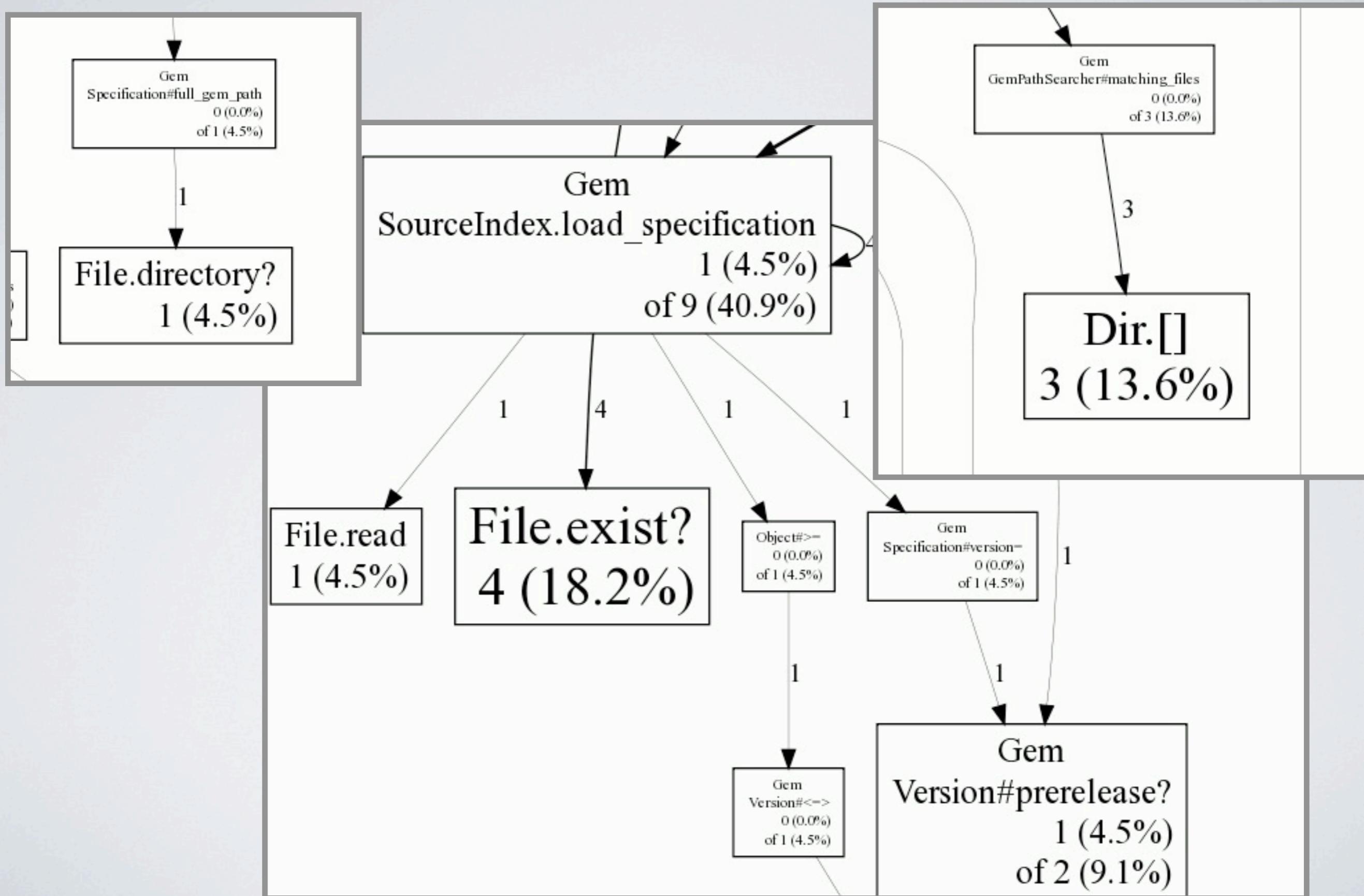
Sinatra  
Application#GET  
/compute  
83 (35.8%)  
of 118 (50.9%)

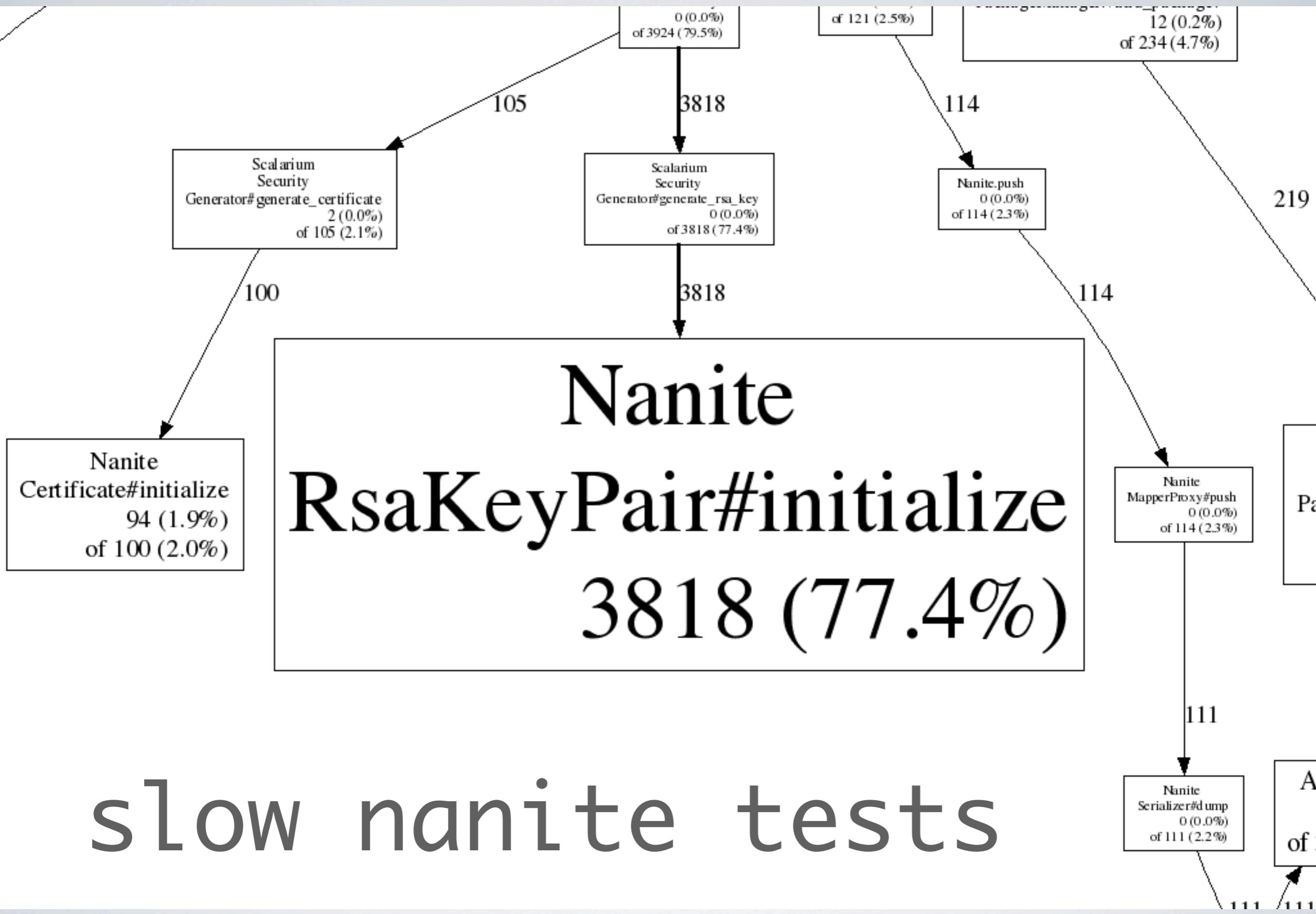


# redis-rb bottleneck



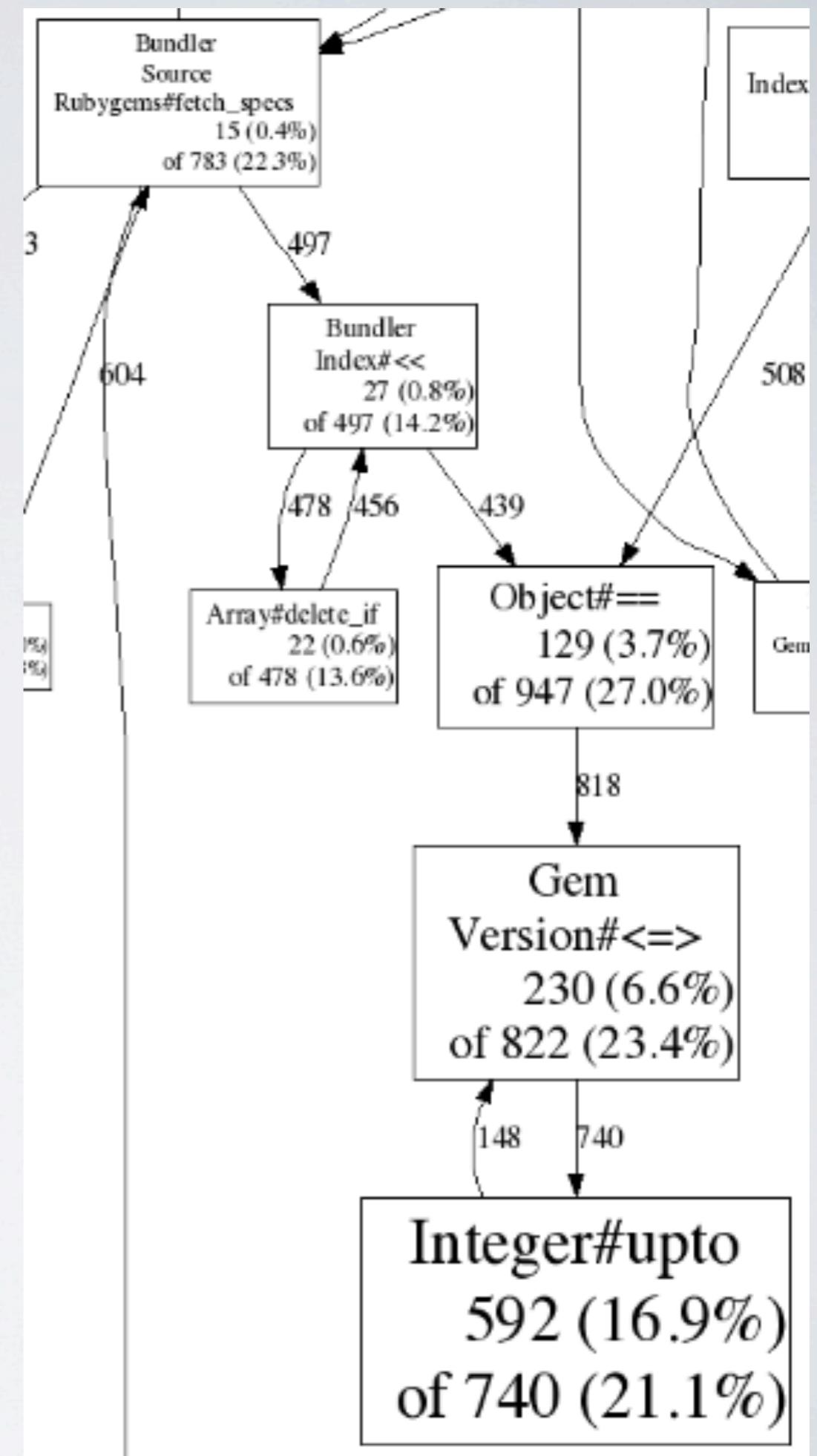
# why is rubygems slow?





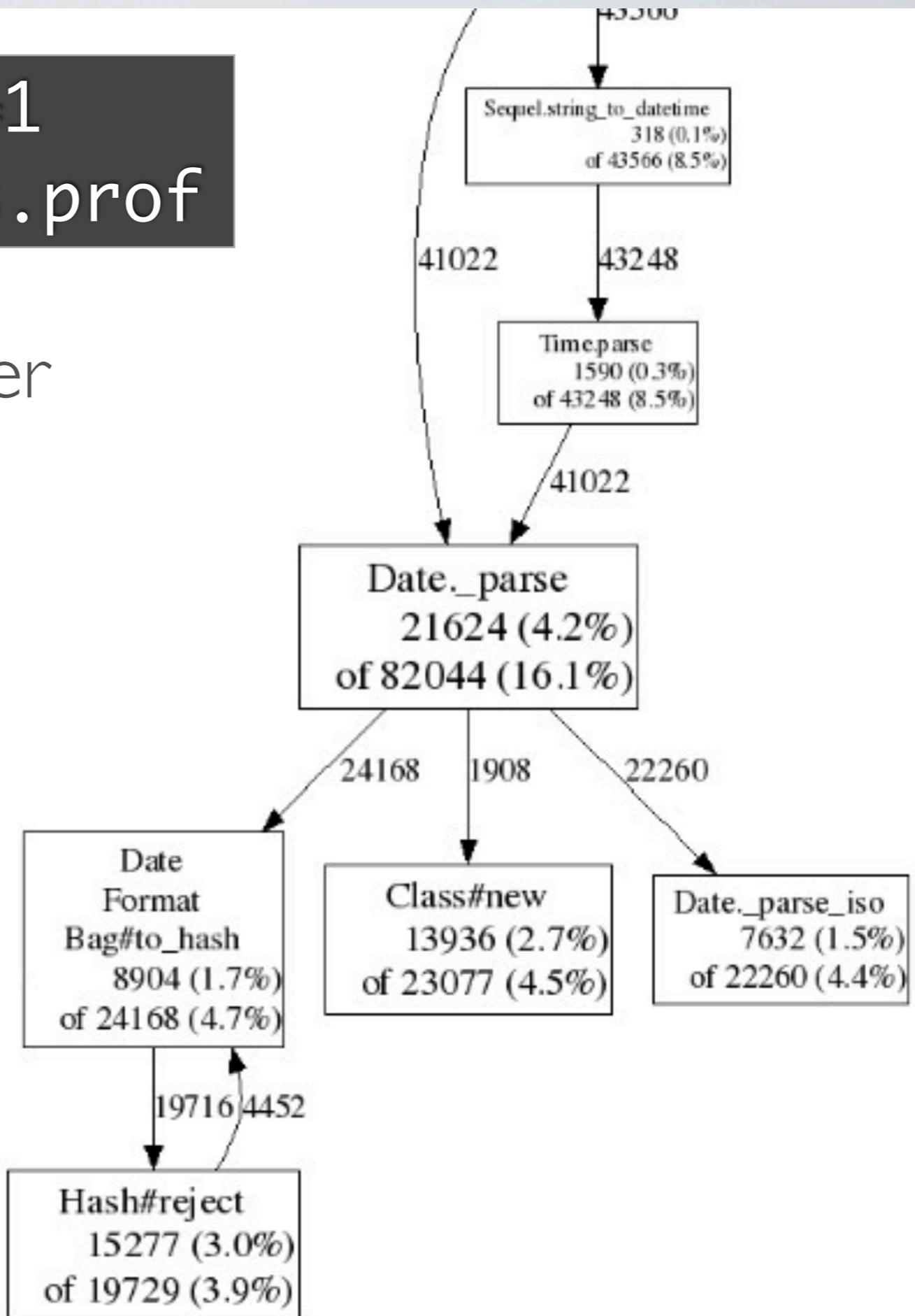
# faster bundle install

- 23% spent in  
`Gem::Version#<=>`
- simple patch to rubygems  
improved overall install  
performance by 15%
- <http://gist.github.com/458185>



CPUPROFILE\_OBJECTS=1  
CPUPROFILE=app-objs.prof

- object allocation profiler mode built-in
- 1 sample = 1 object created
- Time parsing is both CPU and object allocation intensive
- using mysql2 moves this to C



# RACK-PERFTOOLS

rack middleware for perftools.rb

```
gem install rack-  
perftools_profiler
```

[github.com/bhb/rack-perftools\\_profiler](https://github.com/bhb/rack-perftools_profiler)

```
require 'rack/perftools_profiler'  
config.middleware.insert(  
  0,  
  Rack::PerftoolsProfiler,  
  :default_printer => 'gif'  
)
```

```
$ curl http://localhost:3000/_start_  
$ curl http://localhost:3000/home  
$ curl http://localhost:3000/about  
$ curl http://localhost:3000/_stop_  
  
$ curl http://localhost:3000/_data_ -o profile.gif  
$ curl http://localhost:3000/_data_?printer=text -o profile.txt  
  
$ curl "http://localhost:3000/home?profile=true&times=10"  
      -o 10_requests_to_homepage.gif
```

# GDB

## the GNU debugger

```
gdb <executable>
gdb attach <pid>
```

# Debugging Ruby Segfaults

test\_segv.rb:4: [BUG] Segmentation fault  
ruby 1.8.7 (2008-08-11 patchlevel 72) [i686-darwin9.7.0]

```
#include "ruby.h"

VALUE
segv()
{
    VALUE array[1];
    array[100000] = NULL;
    return Qnil;
}

void
Init_segv()
{
    rb_define_method(rb_cObject, "segv", segv, 0);
}
```

# Debugging Ruby Segfaults

```
test_segv.rb:4: [BUG] Segmentation fault  
ruby 1.8.7 (2008-08-11 patchlevel 72) [i686-darwin9.7.0]
```

```
#include "ruby.h"  
  
VALUE  
segv()  
{  
    VALUE array[1];  
    array[100000] = NULL;  
    return Qnil;  
}  
  
void  
Init_segv()  
{  
    rb_define_method(rb_cObject, "segv", segv, 0);  
}
```

```
def test  
  require 'segv'  
  4.times do  
    Dir.chdir '/tmp' do  
      Hash.new{ segv }[0]  
    end  
  end  
end  
  
sleep 10  
test()
```

# 1. Attach to running process

```
$ ps aux | grep ruby  
joe 23611 0.0 0.1 25424 7540 S Dec01 0:00 ruby test_segv.rb
```

```
$ sudo gdb ruby 23611  
Attaching to program: ruby, process 23611  
0x00007fa5113c0c93 in nanosleep () from /lib/libc.so.6  
(gdb) c  
Continuing.
```

```
Program received signal SIGBUS, Bus error.  
segv () at segv.c:7  
7     array[1000000] = NULL;
```

# 1. Attach to running process

```
$ ps aux | grep ruby  
joe 23611 0.0 0.1 25424 7540 S Dec01 0:00 ruby test_segv.rb
```

```
$ sudo gdb ruby 23611  
Attaching to program: ruby, process 23611  
0x00007fa5113c0c93 in nanosleep () from /lib/libc.so.6  
(gdb) c  
Continuing.
```

```
Program received signal SIGBUS, Bus error.  
segv () at segv.c:7  
7     array[1000000] = NULL;
```

# 2. Use a coredump

```
Process.setrlimit Process::RLIMIT_CORE, 300*1024*1024
```

```
$ sudo mkdir /cores  
$ sudo chmod 777 /cores  
$ sudo sysctl kernel.core_pattern=/cores/%e.core.%s.%p.%t  
  
$ sudo gdb ruby /cores/ruby.core.6.23611.1259781224
```

```
def test
  require 'segv'
  4.times do
    Dir.chdir '/tmp' do
      Hash.new{ segv }[0]
    end
  end
end
test()
```

(gdb) where

#0 segv () at segv.c:7

#1 0x000000000041f2be in call\_cfunc () at eval.c:5727

...

#13 0x000000000043ba8c in rb\_hash\_default () at hash.c:521

...

#19 0x000000000043b92a in rb\_hash\_aref () at hash.c:429

...

#26 0x00000000004bb7bc in chdir\_yield () at dir.c:728

#27 0x000000000041d8d7 in rb\_ensure () at eval.c:5528

#28 0x00000000004bb93a in dir\_s\_chdir () at dir.c:816

...

#35 0x000000000041c444 in rb\_yield () at eval.c:5142

#36 0x0000000000450690 in int\_dotimes () at numeric.c:2834

...

#48 0x0000000000412a90 in ruby\_run () at eval.c:1678

#49 0x000000000041014e in main () at main.c:48

# GDB.RB

## gdb with MRI hooks

```
gem install gdb.rb  
gdb.rb <pid>
```

[github.com/tmm1/gdb.rb](https://github.com/tmm1/gdb.rb)

(gdb) ruby eval 1+2

3

(gdb) ruby eval Thread.current  
#<Thread:0x1d630 run>

(gdb) ruby eval Thread.list.size  
8

(gdb) ruby threads list

|         |      |        |                 |                    |
|---------|------|--------|-----------------|--------------------|
| 0x15890 | main | thread | THREAD_STOPPED  | WAIT_JOIN(0x19ef4) |
| 0x19ef4 |      | thread | THREAD_STOPPED  | WAIT_TIME(57.10s)  |
| 0x19e34 |      | thread | THREAD_STOPPED  | WAIT_FD(5)         |
| 0x19dc4 |      | thread | THREAD_STOPPED  | WAIT_NONE          |
| 0x19dc8 |      | thread | THREAD_STOPPED  | WAIT_NONE          |
| 0x19dcc |      | thread | THREAD_STOPPED  | WAIT_NONE          |
| 0x22668 |      | thread | THREAD_STOPPED  | WAIT_NONE          |
| 0x1d630 | curr | thread | THREAD_RUNNABLE | WAIT_NONE          |

| (gdb) ruby objects |         |          |
|--------------------|---------|----------|
| HEAPS              | 8       |          |
| SLOTS              | 1686252 |          |
| LIVE               | 893327  | (52.98%) |
| FREE               | 792925  | (47.02%) |
|                    |         |          |
| scope              | 1641    | (0.18%)  |
| regexp             | 2255    | (0.25%)  |
| data               | 3539    | (0.40%)  |
| class              | 3680    | (0.41%)  |
| hash               | 6196    | (0.69%)  |
| object             | 8785    | (0.98%)  |
| array              | 13850   | (1.55%)  |
| string             | 105350  | (11.79%) |
| node               | 742346  | (83.10%) |

(gdb) ruby objects strings

140 u'lib'

158 u'0'

294 u'\n'

619 u' '

30503 unique strings

3187435 bytes

```
def test
  require 'segv'
  4.times do
    Dir.chdir '/tmp' do
      Hash.new{ segv }[0]
    end
  end
end

test()
```

```
def test
  require 'segv'
  4.times do
    Dir.chdir '/tmp' do
      Hash.new{ segv }[0]
    end
  end
end
```

```
(gdb) ruby threads
test()
```

| 0xa3e000 | main       | curr | thread  | THREAD_RUNNABLE   |
|----------|------------|------|---------|-------------------|
|          | node_vcall |      | segv    | in test_segv.rb:5 |
|          | node_call  |      | test    | in test_segv.rb:5 |
|          | node_call  |      | call    | in test_segv.rb:5 |
|          | node_call  |      | default | in test_segv.rb:5 |
|          | node_call  |      | []      | in test_segv.rb:5 |
|          | node_call  |      | test    | in test_segv.rb:4 |
|          | node_call  |      | chdir   | in test_segv.rb:4 |
|          | node_call  |      | test    | in test_segv.rb:3 |
|          | node_call  |      | times   | in test_segv.rb:3 |
|          | node_vcall |      | test    | in test_segv.rb:9 |

# rails\_warden leak

```
(gdb) ruby objects classes
1197 MIME::Type
2657 NewRelic::MetricSpec
2719 TZInfo::TimezoneTransitionInfo
4124 Warden::Manager
4124 MethodOverrideForAll
4124 AccountMiddleware
4124 Rack::Cookies
4125 ActiveRecord::ConnectionAdapters::ConnectionManagement
4125 ActionController::Session::CookieStore
4125 ActionController::Failsafe
4125 ActionController::ParamsParser
4125 Rack::Lock
4125 ActionController::Dispatcher
4125 ActiveRecord::QueryCache
4125 ActiveSupport::MessageVerifier
4125 Rack::Head
```

middleware chain leaking per request

# mongrel sleeper thread

|            |   |                |                 |            |
|------------|---|----------------|-----------------|------------|
| 0x16814c00 | thread  | THREAD_STOPPED | WAIT_TIME(0.47) | 1522 bytes |
| node_fcall | sleep in lib/mongrel/configurator.rb:285      |                |                 |            |
| node_fcall | run in lib/mongrel/configurator.rb:285        |                |                 |            |
| node_fcall | loop in lib/mongrel/configurator.rb:285       |                |                 |            |
| node_call  | run in lib/mongrel/configurator.rb:285        |                |                 |            |
| node_call  | initialize in lib/mongrel/configurator.rb:285 |                |                 |            |
| node_call  | new in lib/mongrel/configurator.rb:285        |                |                 |            |
| node_call  | run in bin/mongrel_rails:128                  |                |                 |            |
| node_call  | run in lib/mongrel/command.rb:212             |                |                 |            |
| node_call  | run in bin/mongrel_rails:281                  |                |                 |            |
| node_fcall | (unknown) in bin/mongrel_rails:19             |                |                 |            |

```
def run
  @listeners.each {|name,s|
    s.run
  }

  $mongrel_sleeper_thread = Thread.new { loop { sleep 1 } }
end
```

# god memory leaks

```
(gdb) ruby objects arrays  
elements instances
```

```
94310 3
```

```
94311 3
```

```
94314 2
```

```
94316 1
```

```
5369 arrays
```

```
2863364 member elements
```

many arrays with  
90k+ elements!

```
43 God::Process  
43 God::Watch  
43 God::Driver  
43 God::DriverEventQueue  
43 God::Conditions::MemoryUsage  
43 God::Conditions::ProcessRunning  
43 God::Behaviors::CleanPidFile  
45 Process::Status  
86 God::Metric  
327 God::System::SlashProcPoller  
327 God::System::Process  
406 God::DriverEvent
```

5 separate god leaks fixed by Eric  
Lindvall with the help of gdb.rb!

# MEMPROF

a heap visualizer for ruby

```
gem install memprof  
open http://memprof.com
```

[github.com/ice799/memprof](https://github.com/ice799/memprof)

# memprof features

- memprof.track
- memprof.dump
- memprof.dump\_all
- memprof.com



```
Memprof.track{  
  100.times{ "abc" }  
  100.times{ 1.23 + 1 }  
  100.times{ Module.new }  
}
```

```
100    file.rb:2:String  
100    file.rb:3:Float  
100    file.rb:4:Module
```

- like break\_house, but for a given block of code
- use `Memprof::Middleware` in your webapps to run track per request

# memprof features

- memprof.track
- memprof.dump
- memprof.dump\_all
- memprof.com



# strings

```
Memprof.dump{  
    "hello" + "world"  
}
```

# strings

```
{  
  "_id": "0x19c610",  
  "file": "file.rb",  
  "line": 2,  
  
  "type": "string",  
  "class": "0x1ba7f0",  
  "class_name": "String",  
  
  "length": 10,  
  "data": "helloworld"  
}
```

```
Memprof.dump{  
  "hello" + "world"  
}
```

← memory address of object

# strings

```
{  
  "_id": "0x19c610",  
  "file": "file.rb",  
  "line": 2,  
  
  "type": "string",  
  "class": "0x1ba7f0",  
  "class_name": "String",  
  
  "length": 10,  
  "data": "helloworld"  
}
```

```
Memprof.dump{  
  "hello" + "world"  
}
```

← memory address of object  
← file and line where string was created

# strings

```
{  
  "_id": "0x19c610",  
  "file": "file.rb",  
  "line": 2,  
  
  "type": "string",  
  "class": "0x1ba7f0",  
  "class_name": "String",  
  
  "length": 10,  
  "data": "helloworld"  
}
```

```
Memprof.dump{  
  "hello" + "world"  
}
```

← memory address of object  
← file and line where string was created  
← address of the class “String”

# strings

```
{  
  "_id": "0x19c610",  
  "file": "file.rb",  
  "line": 2,  
  
  "type": "string",  
  "class": "0x1ba7f0",  
  "class_name": "String",  
  
  "length": 10,  
  "data": "helloworld"  
}
```

← memory address of object

← file and line where string was created

← address of the class “String”

← length and contents of this string instance

```
Memprof.dump{  
  "hello" + "world"  
}
```

# arrays

```
Memprof.dump{  
  [  
    1,  
    :b,  
    2.2,  
    "d"  
  ]  
}
```

# arrays

```
{  
  "_id": "0x19c5c0",  
  "class": "0x1b0d18",  
  "class_name": "Array",  
  
  "length": 4,  
  "data": [  
    1,  
    ":b",  
  
    "0x19c750",  
    "0x19c598"  
  ]  
}
```

```
Memprof.dump{  
  [  
    1,  
    ":b",  
    2.2,  
    "d"  
  ]  
}
```

# arrays

```
{  
  "_id": "0x19c5c0",  
  "class": "0x1b0d18",  
  "class_name": "Array",  
  
  "length": 4,  
  "data": [  
    1,  
    ":b",  
    "0x19c750",  
    "0x19c598"  
  ]  
}
```

```
Memprof.dump{  
  [  
    1,  
    ":b",  
    2.2,  
    "d"  
  ]  
}
```

← integers and symbols are stored in the array itself

# arrays

```
{  
  "_id": "0x19c5c0",  
  "class": "0x1b0d18",  
  "class_name": "Array",  
  
  "length": 4,  
  "data": [  
    1,  
    ":b",  
    "0x19c750",  
    "0x19c598"  
  ]  
}
```

```
Memprof.dump{  
  [  
    1,  
    :b,  
    2.2,  
    "d"  
  ]  
}
```

→ integers and symbols are stored in the array itself

→ floats and strings are separate ruby objects

# hashes

```
Memprof.dump{  
    {  
        :a => 1,  
        "b" => 2.2  
    }  
}
```

# hashes

```
{  
  "_id": "0x19c598",  
  
  "type": "hash",  
  "class": "0x1af170",  
  "class_name": "Hash",  
  
  "default": null,  
  
  "length": 2,  
  "data": [  
    [ ":a", 1 ],  
    [ "0xc728", "0xc750" ]  
  ]  
}
```

```
Memprof.dump{  
  :a => 1,  
  "b" => 2.2  
}
```

# hashes

```
{  
  "_id": "0x19c598",  
  
  "type": "hash",  
  "class": "0x1af170",  
  "class_name": "Hash",  
  
  "default": null,  
  
  "length": 2,  
  "data": [  
    [ ":a", 1 ],  
    [ "0xc728", "0xc750" ]  
  ]  
}
```

hash entries as key/value  
pairs

```
Memprof.dump{  
  :a => 1,  
  "b" => 2.2  
}
```

# hashes

```
{  
  "_id": "0x19c598",  
  
  "type": "hash",  
  "class": "0x1af170",  
  "class_name": "Hash",  
  
  "default": null,           ← no default proc  
  
  "length": 2,  
  "data": [  
    [ ":a", 1 ],  
    [ "0xc728", "0xc750" ]   ← hash entries as key/value  
  ]  
}
```

```
Memprof.dump{  
  :a => 1,  
  "b" => 2.2  
}
```

# classes

```
Memprof.dump{
    class Hello
        @@var=1
        Const=2
        def world() end
    end
}
```

# classes

```
{  
  "_id": "0x19c408",  
  
  "type": "class",  
  "name": "Hello",  
  "super": "0xbfa48",  
  "super_name": "Object",  
  
  "ivars": {  
    "@@var": 1,  
    "Const": 2  
  },  
  "methods": {  
    "world": "0x19c318"  
  }  
}
```

```
Memprof.dump{  
  class Hello  
    @@var=1  
    Const=2  
    def world() end  
  end  
}
```

# classes

```
{  
  "_id": "0x19c408",  
  
  "type": "class",  
  "name": "Hello",  
  "super": "0xbfa48", ← superclass object reference  
  "super_name": "Object",  
  
  "ivars": {  
    "@@var": 1,  
    "Const": 2  
  },  
  "methods": {  
    "world": "0x19c318"  
  }  
}
```

```
Memprof.dump{  
  class Hello  
    @@var=1  
    Const=2  
    def world() end  
  end  
}
```

← superclass object reference

# classes

```
{  
  "_id": "0x19c408",  
  
  "type": "class",  
  "name": "Hello",  
  "super": "0x1bfa48", ← superclass object reference  
  "super_name": "Object",  
  
  "ivars": {  
    "@@var": 1,  
    "Const": 2  
  }, ← class variables and constants  
  "methods": {  
    "world": "0x19c318"  
  }  
}
```

```
Memprof.dump{  
  class Hello  
    @@var=1  
    Const=2  
    def world() end  
  end  
}
```

← are stored in the instance variable table

# classes

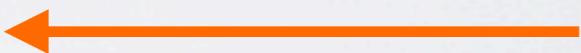
```
{  
  "_id": "0x19c408",  
  
  "type": "class",  
  "name": "Hello",  
  "super": "0x1bfa48", ← superclass object reference  
  "super_name": "Object",  
  
  "ivars": {  
    "@@var": 1,  
    "Const": 2  
  }, ← class variables and constants  
  "methods": {  
    "world": "0x19c318" ← references to method objects  
  }  
}
```

```
Memprof.dump{  
  class Hello  
    @@var=1  
    Const=2  
    def world() end  
  end  
}
```

← are stored in the instance variable table

# memprof features

- memprof.track
- memprof.dump
- memprof.dump\_all
- memprof.com



```
Memprof.dump_all("myapp_heap.json")
```

- dump out every single live object as json
- one per line to specified file
- analyze via
  - jsawk/grep
  - mongodb/couchdb
  - custom ruby scripts
  - libyajl + Boost Graph Library

# memprof features

- memprof.track
- memprof.dump
- memprof.dump\_all
- memprof.com



# memprof.com

a web-based heap visualizer and leak analyzer

## **new rails3-beta application** by **tmm1** about a month ago

### **ruby-1.8.7-p249/bin/ruby**

- ruby 1.8.7 (2010-01-10 patchlevel 249) [i686-darwin10.2.0]
- executing ./script/rails
- compiled with -g -O2 -fno-common -pipe -fno-common \$(cflags)
- memory usage is 97156 bytes
- working directory is test/code/newapp
- 6 IO objects and 10 file descriptors
- 20 shared libraries

### **404869 objects**

- 78 global variables
- 213 constants inside Object
- objects grouped by age
- objects grouped by type
- objects with most outbound references

### **2428 classes and 695 modules**

- namespace hierarchy
- class hierarchy
- instances per class
- duplicate classes by name

# plugging a leak in rails3

- in dev mode, rails3 is leaking 10mb per request

| PID  | Process Name | User | % CPU | Threads | Real Mem | Kind           | Virtual Mem |
|------|--------------|------|-------|---------|----------|----------------|-------------|
| 3296 | ruby         | test | 0.1   | 1       | 536.3 MB | Intel (64 bit) | 625.6 MB    |

# plugging a leak in rails3

- in dev mode, rails3 is leaking 10mb per request

| PID  | Process Name | User | % CPU | Threads | Real Mem | Kind           | Virtual Mem |
|------|--------------|------|-------|---------|----------|----------------|-------------|
| 3296 | ruby         | test | 0.1   | 1       | 536.3 MB | Intel (64 bit) | 625.6 MB    |

- ✿ let's use memprof to find it!

```
# in environment.rb  
require `gem which memprof/signal`.strip
```

# plugging a leak in rails3

- ✿ send the app some requests so it leaks

```
$ ab -c 1 -n 30  
http://localhost:3000/
```

- ✿ tell memprof to dump out the entire heap to json

```
$ memprof  
  --pid <pid>  
  --name <dump name>  
  --key <api key>
```

# plugging a leak in rails3

- ✿ send the app some requests so it leaks

```
$ ab -c 1 -n 30
http://localhost:3000/
```

- ✿ tell memprof to dump out the entire heap to json

```
$ memprof
--pid <pid>
--name <dump name>
--key <api key>
```

**new rails3 app by tmm1** 13 minutes ago

## **ruby-1.8.7-p249/bin/ruby**

- ruby 1.8.7 (2010-01-10 patchlevel 249) [i686-darwin10.2.0]
- executing ./script/rails
- compiled with -g -O2 -fno-common -pipe -fno-common \$(cflags)
- memory usage is 141080 bytes
- working directory is test/code/newapp
- 6 IO objects and 10 file descriptors
- 20 shared libraries

## **443586 objects**

- 78 global variables
- 214 constants inside Object
- 30 finalizers
- objects grouped by age
- objects grouped by type
- objects with most outbound references

## **2519 classes and 777 modules**

- namespace hierarchy
- class hierarchy
- instances per class
- duplicate classes by name

## **892 source files**

- 1977 objects created inside eval
- 702 objects missing file information

## **315149 nodes**

- nodes by type
- nodes per file

## **12798 methods**

- methods per file

## **12233 arrays**

- 15 subclasses of Array
- largest array has 897 elements
- group by size

**2519 classes**



```
{"type": "class"}  
  
2519 objects list group by name ↴  
  
1483 +·(unknown)  
30 +· ApplicationController  
30 -· TestController  
30 +·newapp/app/controllers/test_controller.rb
```

2519 classes

30 copies of  
TestController

2519 objects list group by name

- 1483 +·(unknown)
- 30 +·ApplicationController
- 30 -·TestController
- 30 +·newapp/app/controllers/test\_controller.rb

2519 classes

30 copies of  
TestController

```
{"type": "class"}
```

2519 objects list group by name ↴

1483 + (unknown)

30 + ApplicationController

30 - TestController

30 + newapp/app/controllers/test\_controller.rb

```
{"name": "TestController",  
 "type": "class"}
```

30 objects list group ⚙

0x1115380 TestController

0x16061a0 TestController

0x211d2a0 TestController

0x21f86e8 TestController

0x22240e0 TestController

0x4234ec0 TestController

0x42d5898 TestController

0x4580fe8 TestController

0x4586718 TestController

mongo query for all  
TestController classes

2519 classes

30 copies of  
TestController

```
{"type": "class"}
```

2519 objects list group by name ↴

1483 + (unknown)

30 + ApplicationController

30 - TestController

30 + newapp/app/controllers/test\_controller.rb

```
{"name": "TestController",  
 "type": "class"}
```

30 objects list group ↴

0x1115380 TestController

0x16061a0 TestController

0x211d2a0 TestController

0x21f86e8 TestController

0x22240e0 TestController

0x4234ec0 TestController

0x42d5898 TestController

0x4580fe8 TestController

0x4586718 TestController

mongo query for all  
TestController classes

details for one copy of  
TestController

1 object detail subclasses references

[-] TestController

- [+] node:CREF (TestController)
- [+] #<TestController:0x2137718>
- [+] node:CREF (TestController)
- [+] node:CREF (TestController)
- [+] #<Proc:0x21f2db0 active\_support/core\_ext/class
- [+] #<Proc:0x21f7ab8 active\_support/core\_ext/class
- [+] #<MetaClass:TestController>
- [-] #<Hash:0x23e5f28 length=30>
  - [+] ActionView::Partials::PartialRenderer

← find references to object

1 object detail subclasses references

└ TestController

- + node:CREF (TestController)
- + #<TestController:0x2137718>
- + node:CREF (TestController)
- + node:CREF (TestController)
- + #<Proc:0x21f2db0 active\_support/core\_ext/class
- + #<Proc:0x21f7ab8 active\_support/core\_ext/class
- + #<MetaClass:TestController>
- #<Hash:0x23e5f28 length=30>
  - + ActionView::Partials::PartialRenderer

find references to object

1 object detail references

address #<Hash:0x23e5f28 length=30>  
file lib/action\_view/render/partials.rb  
line 178  
type hash  
class Hash  
length 30  
default #<Proc:0x23e5f00  
lib/action\_view/render/partials.rb:178  
>

TestController #<Hash:0x8df0788 length=0>  
TestController #<Hash:0x325d10 length=0>  
TestController #<Hash:0x16106c8 length=0>  
TestController #<Hash:0x20deeb0 length=0>

1 object detail subclasses references

└ TestController

- + node:CREF (TestController)
- + #<TestController:0x2137718>
- + node:CREF (TestController)
- + node:CREF (TestController)
- + #<Proc:0x21f2db0 active\_support/core\_ext/class
- + #<Proc:0x21f7ab8 active\_support/core\_ext/class
- + #<MetaClass:TestController>
- #<Hash:0x23e5f28 length=30>

+ ActionView::Partials::PartialRenderer

find references to object

1 object detail references

address #<Hash:0x23e5f28 length=30>  
file lib/action\_view/render/partials.rb  
line 178  
type hash  
class Hash  
length 30  
default #<Proc:0x23e5f00  
lib/action\_view/render/partials.rb:178  
>

“leak” is on line 178

holding references to all controllers

TestController #<Hash:0x8df0788 length=0>  
TestController #<Hash:0x325d10 length=0>  
TestController #<Hash:0x16106c8 length=0>  
TestController #<Hash:0x20deeb0 length=0>

- In development mode, Rails reloads all your application code on every request
- ActionView::Partials::PartialRenderer is caching partials used by each controller as an optimization
- But.. it ends up holding a reference to every single reloaded version of those controllers

```
--- a/actionpack/lib/action_view/render/partials.rb
+++ b/actionpack/lib/action_view/render/partials.rb
@@ -179,7 +179,7 @@ module ActionView

    def initialize(view_context, options, block)
        @view           = view_context
-       @partial_names = PARTIAL_NAMES[@view.controller.class]
+       @partial_names = PARTIAL_NAMES[@view.controller.class.name]

        setup(options, block)
    end
```

# HOTSPOTS

a performance tool for rails

```
gem install hotspots  
hotspots /tmp/*.json report/
```

[github.com/tmm1/hotspots](https://github.com/tmm1/hotspots)

```
config.middleware.use(Memprof::Tracer)
```

```
{
```

```
  "time": 4.3442,           ← total time for request
```

```
  "rails": {                ← rails controller/action
```

```
    "controller": "test",
```

```
    "action": "index"
```

```
,
```

```
  "request": {               ← request env info
```

```
    "REQUEST_PATH": "/test",
```

```
    "REQUEST_METHOD": "GET"
```

```
,
```

```
config.middleware.use(Memprof::Tracer)
```

```
"mysql": {  
    "queries": 3,           ← 3 mysql queries  
    "time": 0.00109302  
},
```

```
"gc": {  
    "calls": 8,            ← 8 calls to GC  
    "time": 2.04925        ← 2 secs spent in GC  
},
```

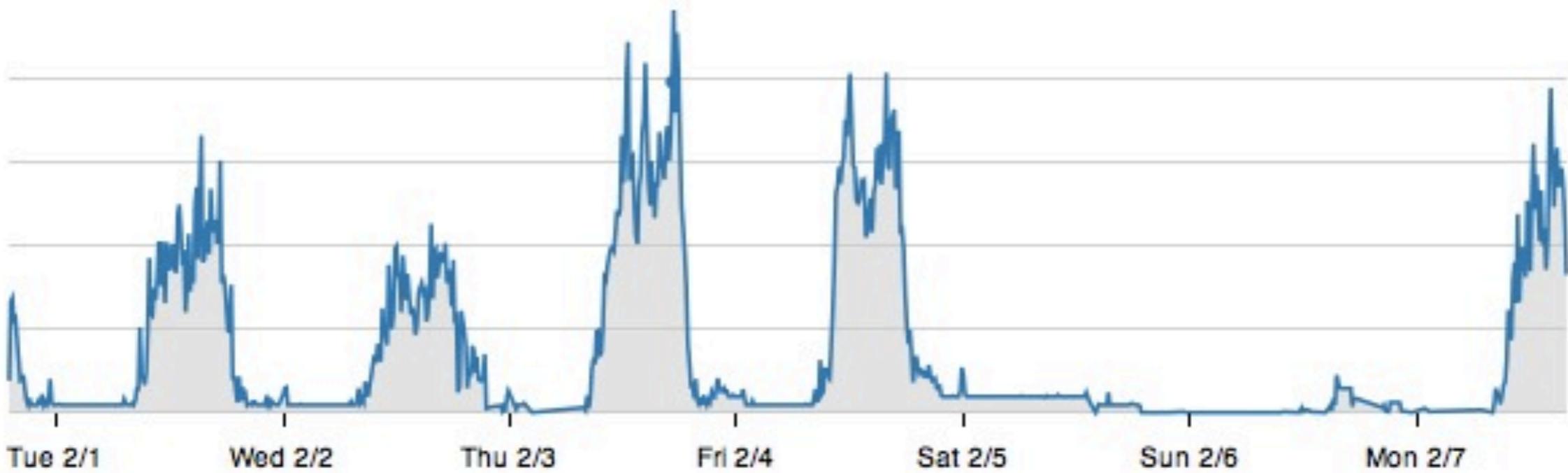
```
config.middleware.use(Memprof::Tracer)

"objects": {
  "created": 3911103, ← 3 million objs created
  "types": {
    "none": 1168831, ← 1 million method calls
    "object": 1127, ← object instances
    "float": 627,
    "string": 1334637, ← lots of strings
    "array": 609313, ← lots of arrays
    "hash": 3676,
    "match": 70211 ← regexp matches
  }
}
}
```

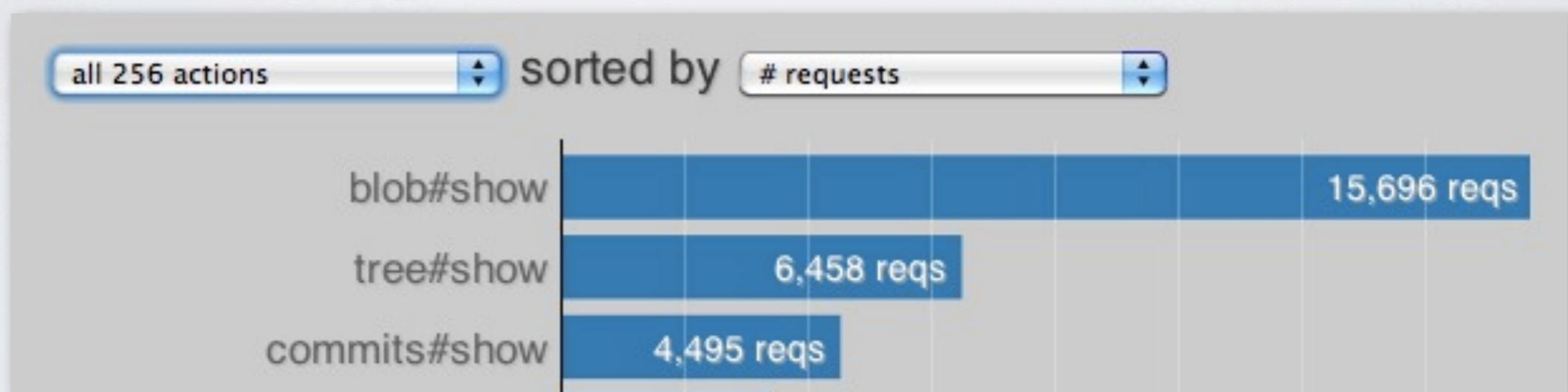
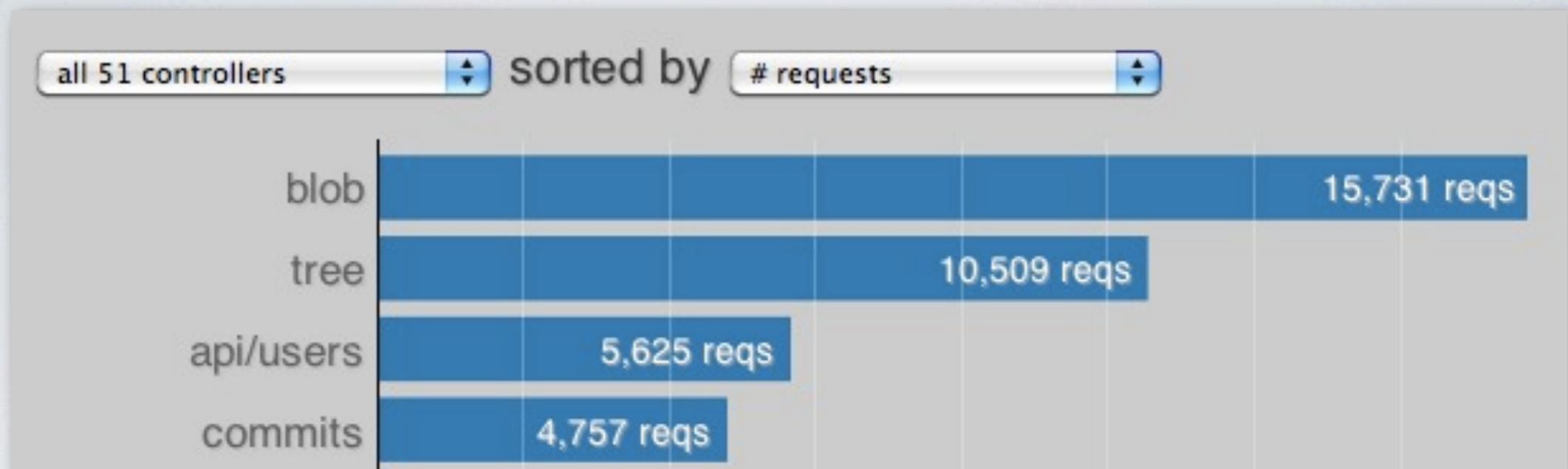
## SUMMARY

**81,427 requests total**

Thu 2/3 5:10pm: 397 reqs (0.7/s)

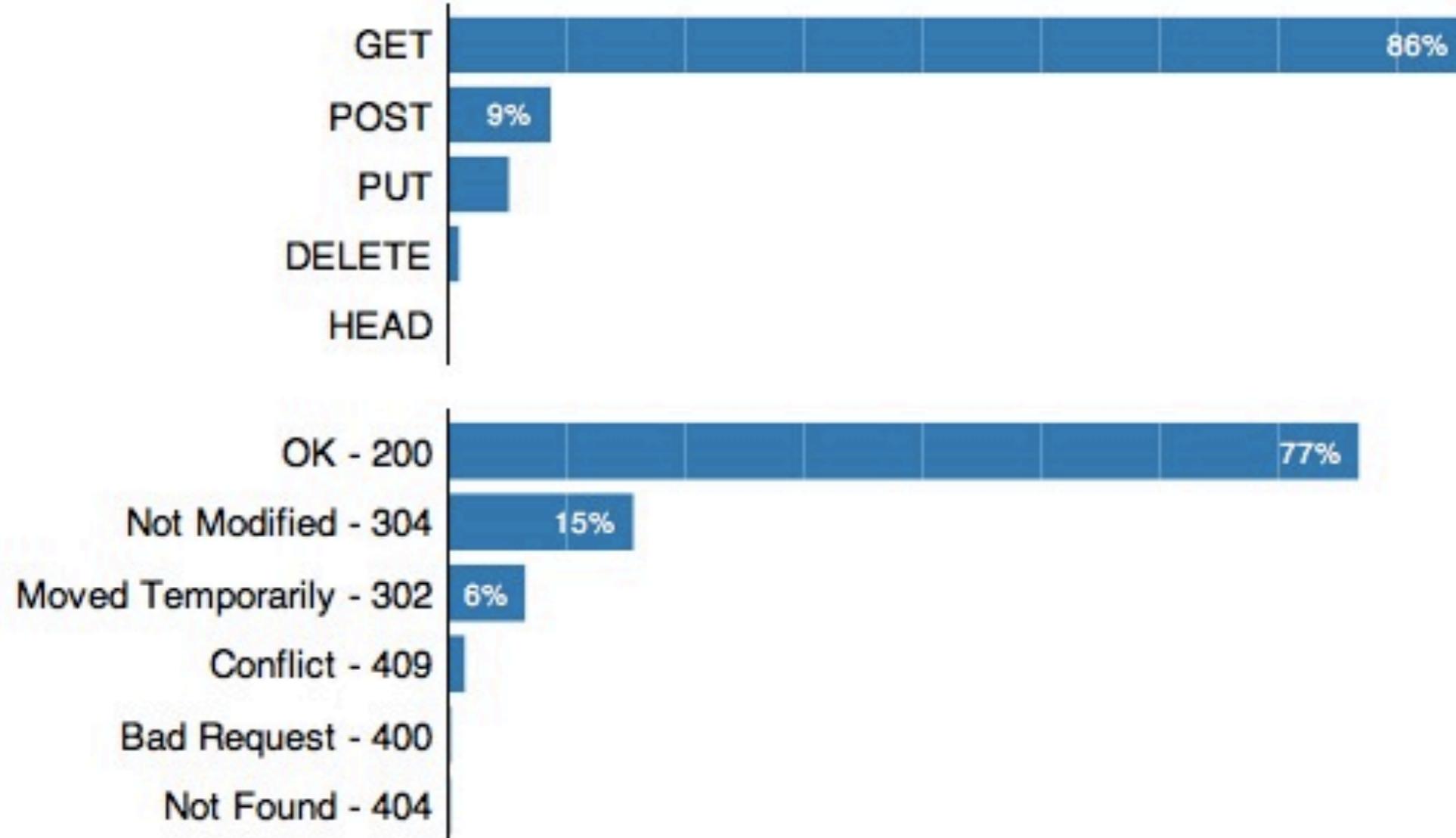


Requests over time



# Top controllers and actions

## REQUEST/RESPONSE BREAKDOWN

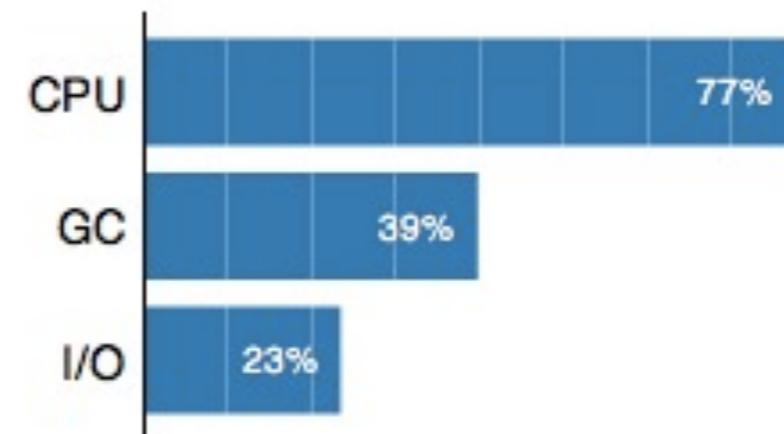
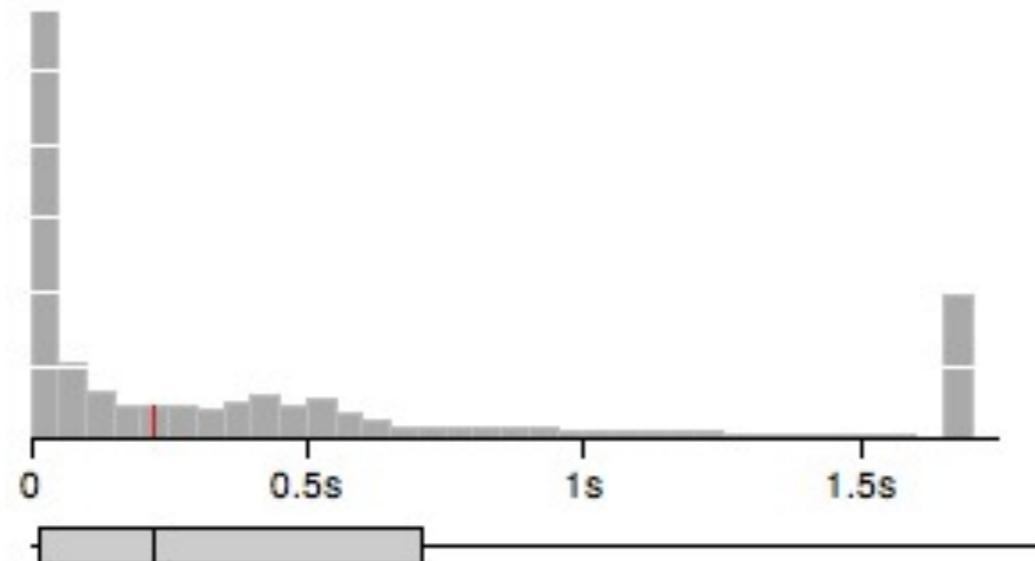


Request and response types

## RESPONSE TIME

**221ms median response time**

*806ms avg / 1110s max*



# Response time breakdown

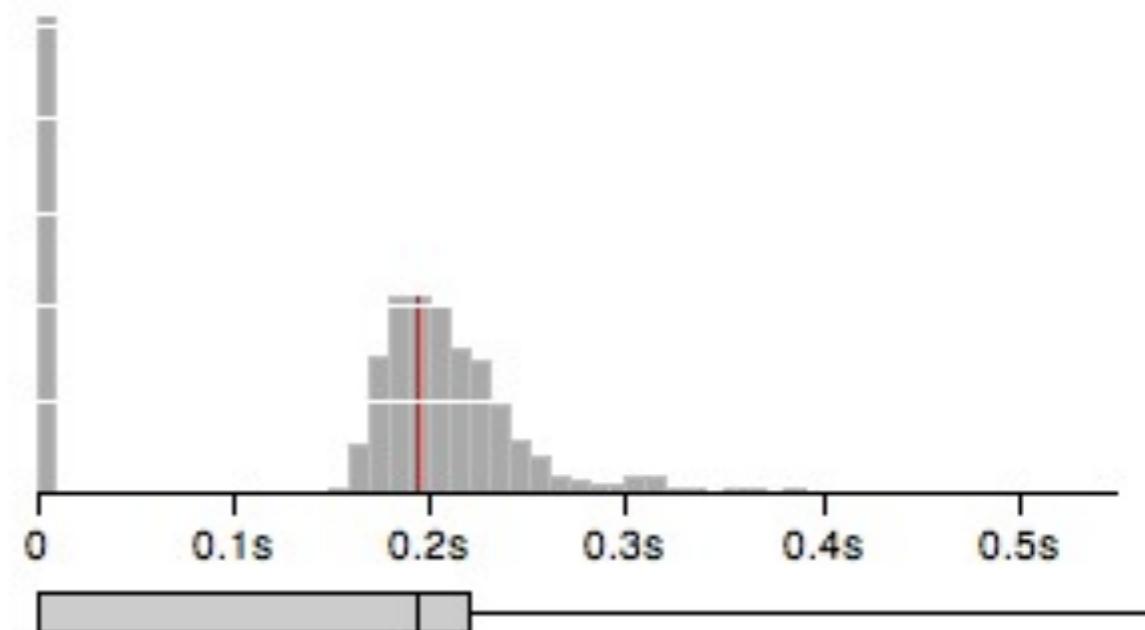
## GARBAGE COLLECTION

**193ms median GC time**

*159ms avg / 1.7s max*

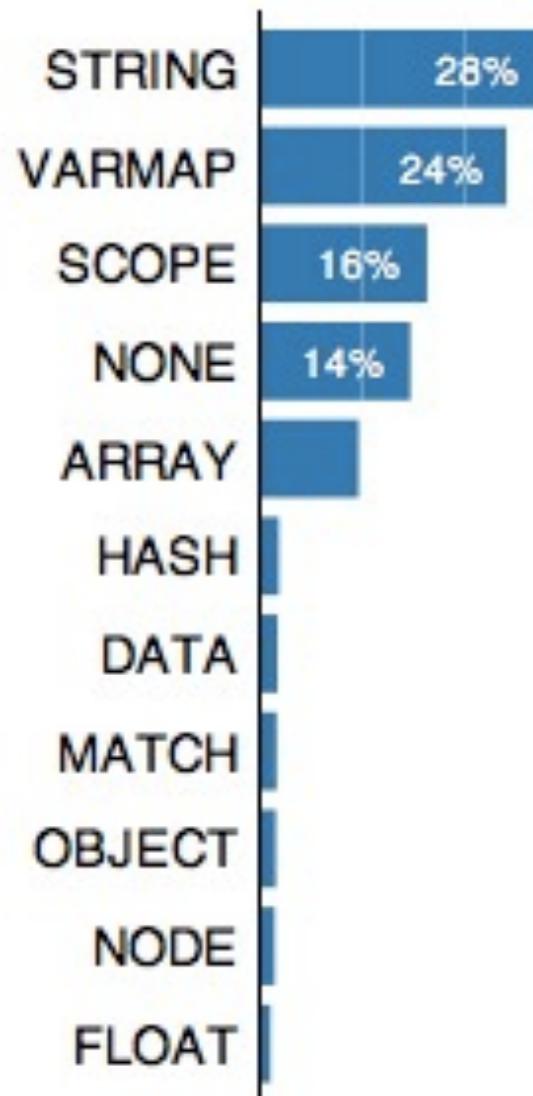
**74% of requests trigger GC**

*2 collections per request / 514 max*



# Garbage Collection overhead

## OBJECTS CREATED



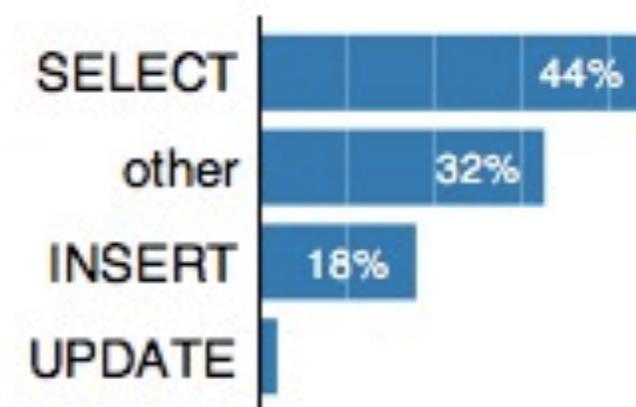
**323,968 objects per request**  
648,222 avg / 190,073,918 max

Objects created per request

## SQL QUERIES

**56 median queries per request**

*220 avg / 21,186 max*



# SQL queries and types



TOOLS MAKE  
DEBUGGING EASIER.



LEARN THESE TOOLS.  
USE THESE TOOLS.

# QUESTIONS?

Aman Gupta  
@tmm1

[speakerdeck.com/u/tmm1/p/debugging-ruby-performance](https://speakerdeck.com/u/tmm1/p/debugging-ruby-performance)