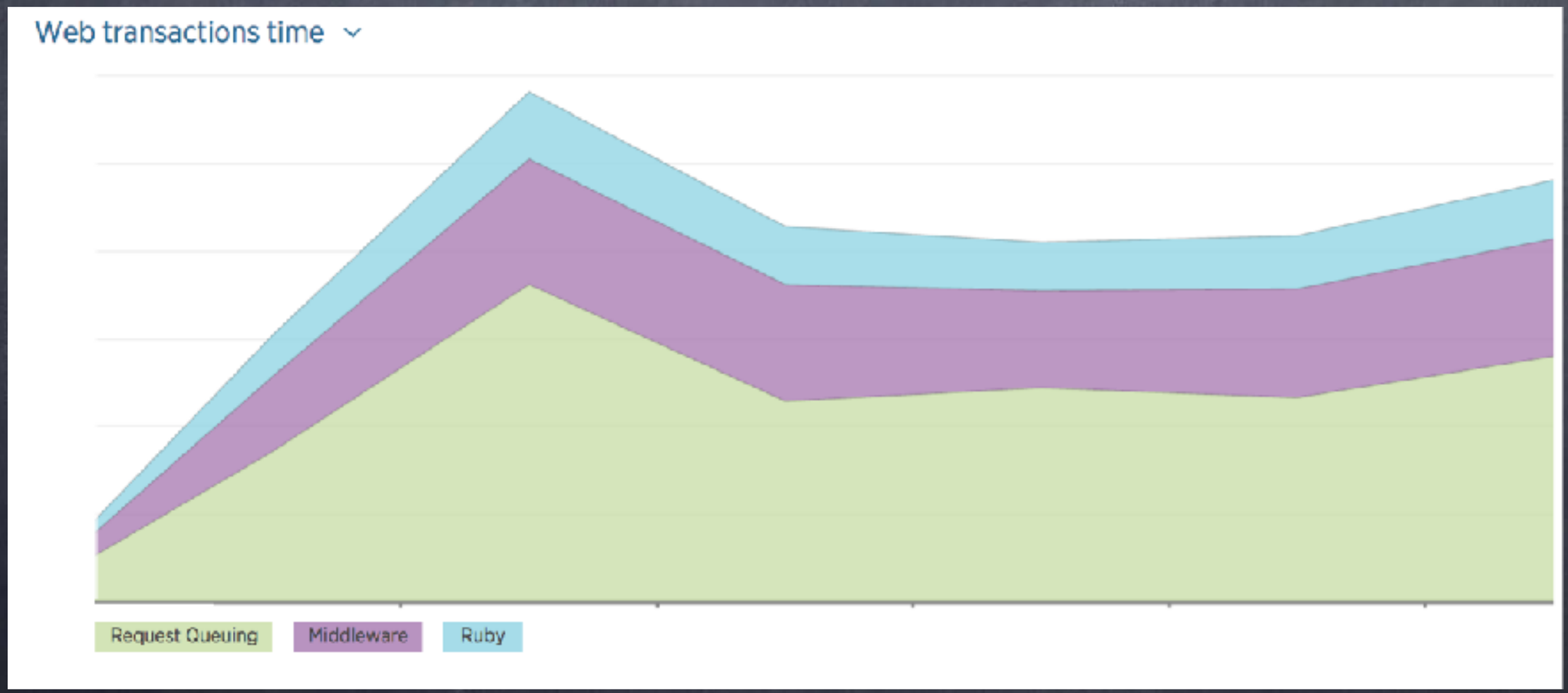
Ruby API Survival Guide

How to survive your first day in production

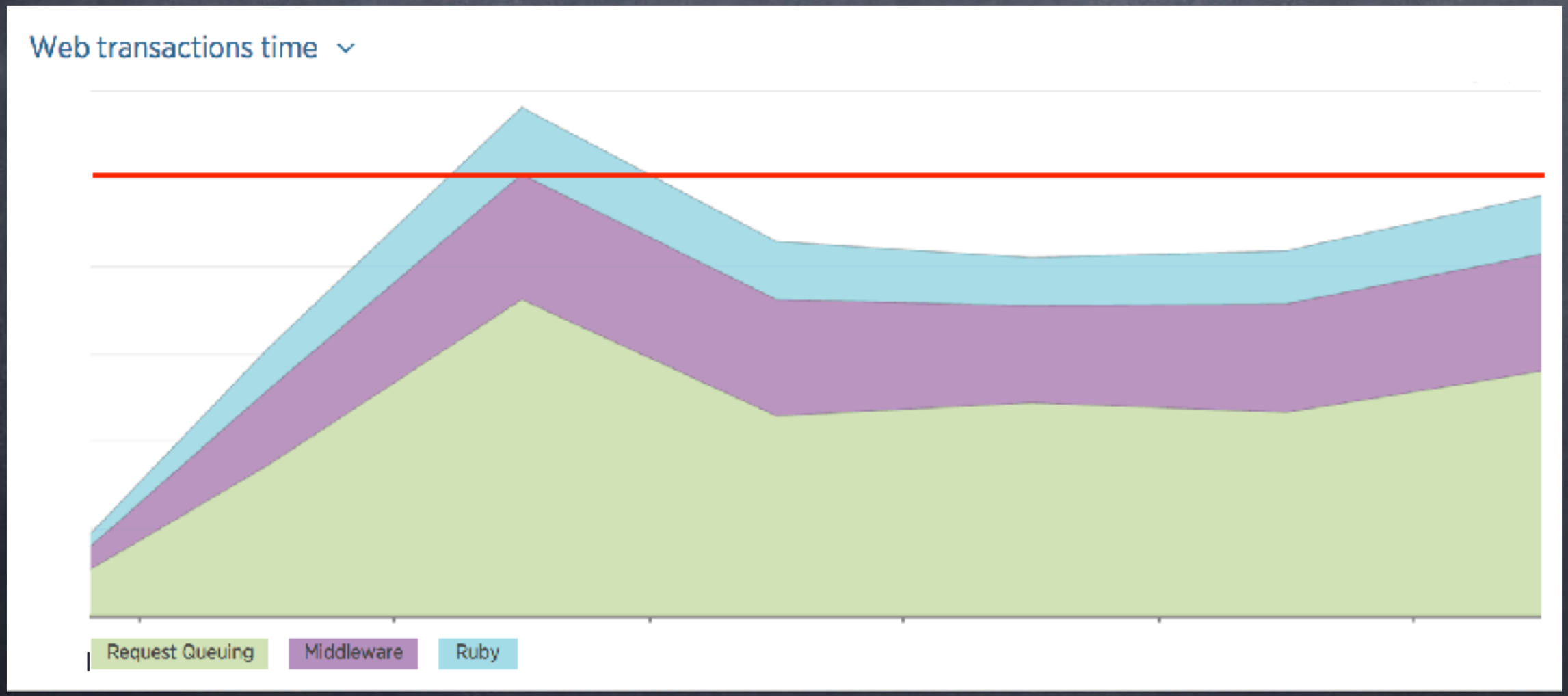
by Miguel Teixeira & Bruno Coelho

2 weeks before going live ...

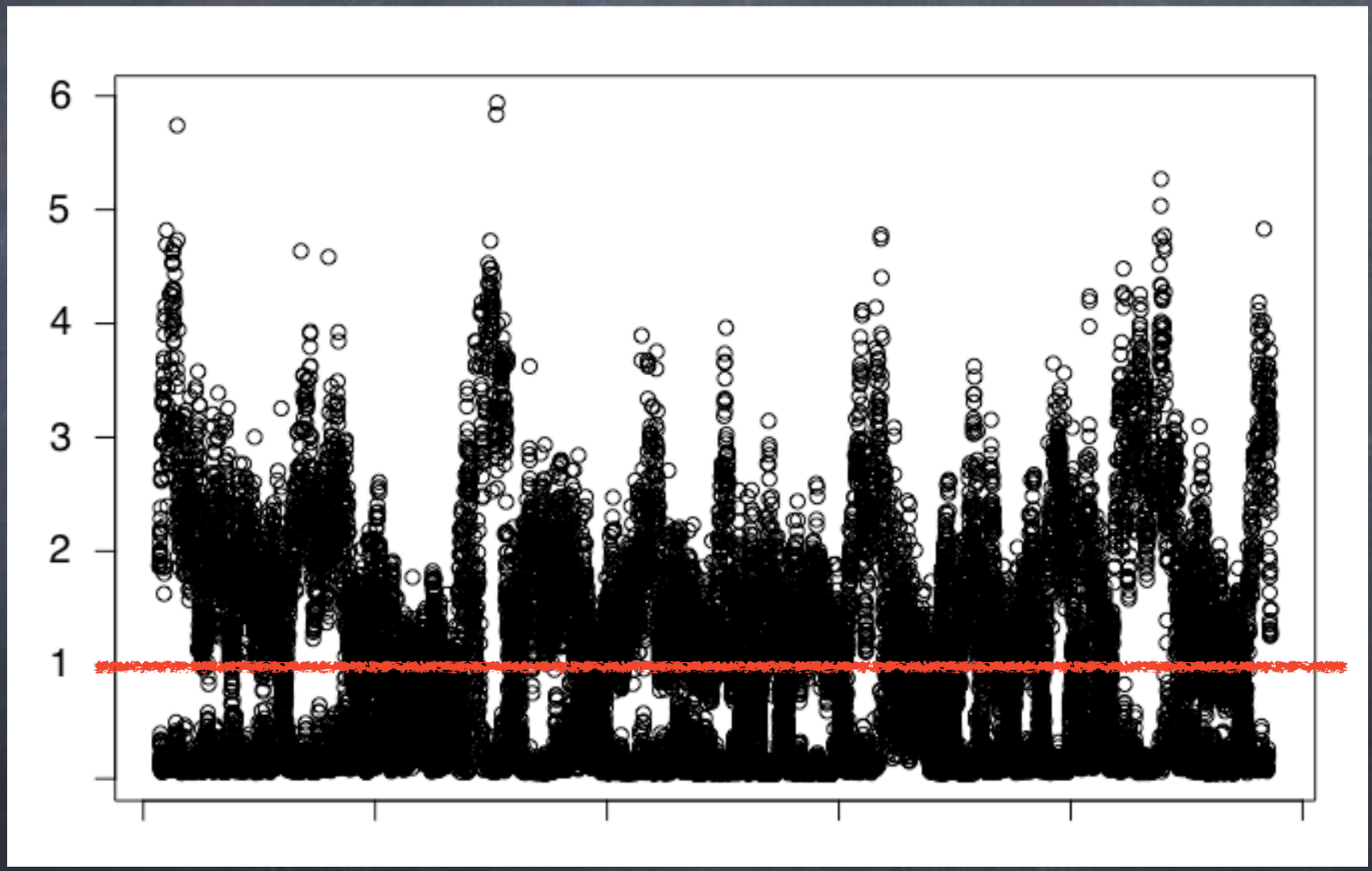
All good!



Watch the red line!



Panic! 😱



💪💪💪

" We can do this.
We know what we are doing! "

WEBrick

- single-process
- single-threaded
- bad for slow connections

Thin

- event-driven
- single-process
- good for slow connections

Unicorn

- worker pool (multi-process)
- single-threaded
- vulnerable to slow clients

Puma

- single-process
- multi-threaded
- good for slow clients, slow I/O

" ... just add more threads,
it's Puma we are using! "

First chapter

Tune your app

Tune your app

- choose your app server
- choose number of processes/workers
- choose number of threads
- size of connection pool
- `preload_app!`
- ...

Second chapter

Optimal number of workers (processes) and threads

workers ?

Container Size - Master Worker Process

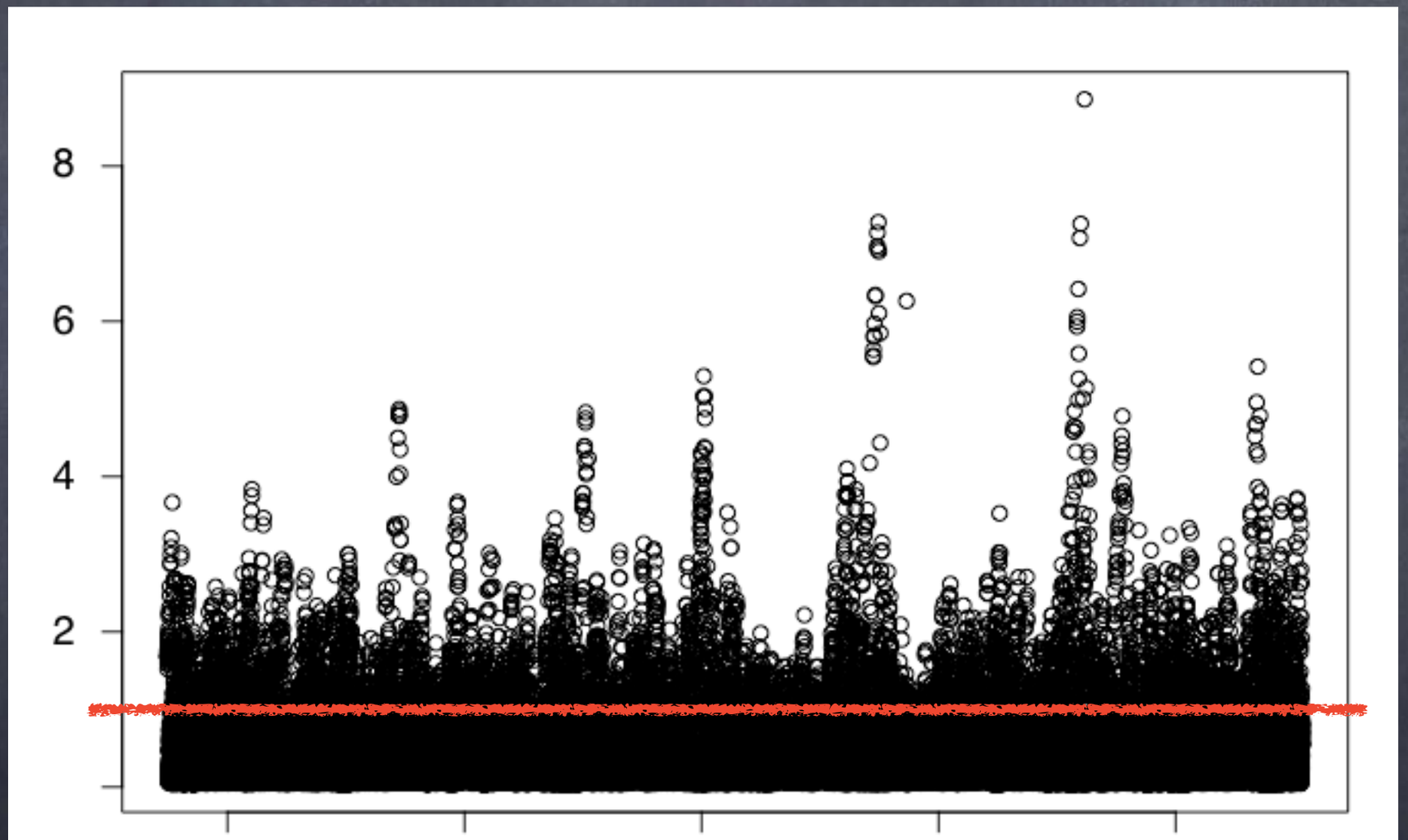
---

1 Worker Process

threads ?

- GIL on MRI
- I/O
- request queuing
- Heroku

Result



Third chapter

Profile your application

ruby-prof

- easy to use
- fast
- configurable
- builtin reports

Call stack

Call tree for application **primes**
Generated on 2015-04-24 11:52:24 +0200 with options {:application=>"primes"}

Threshold: 1.0 Apply Expand All Collapse All Show Help

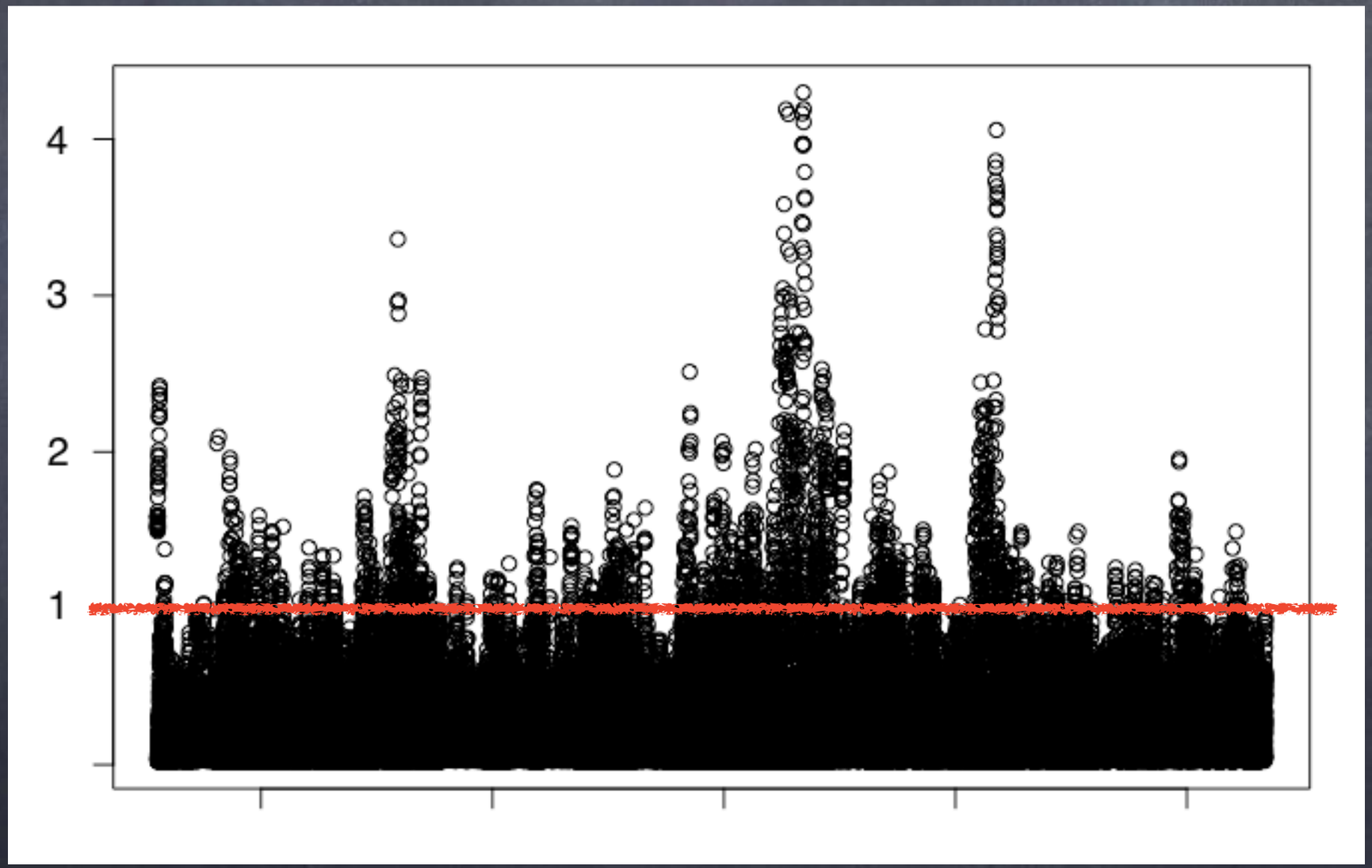
Thread: 70140045951280, Fiber: 70140054192180 (100.00% ~ 0.0029680728912353516)

- 100.00% (100.00%) PrintersTest#setup [1 calls, 1 total]
  - 99.82% (99.82%) Object#run_primes [1 calls, 1 total]
    - 92.62% (92.78%) Object#find_primes [1 calls, 1 total]
      - 92.55% (99.93%) Array#select [1 calls, 1 total]
    - 6.64% (6.65%) Object#make_random_array [1 calls, 1 total]

benchmark-ips

- number of iterations per second
- configurable warm up phase
- easy to read report

Result



Wrapping up

Action points

- make your app tunnable
  - LOAD TEST YOUR APP !
- choose # process & # threads
  - AGAIN ... LOAD TEST YOUR APP !
- profile your app
  - AGAIN & AGAIN ... LOAD TEST YOUR APP !

Thank you!

- Availability team for always being available!
- DevOps for providing us numerous of burning machines 🔥
- and our own awesome team!

Add to Pocket:

- Deploying Rails Applications with the Puma Web Server (https://devcenter.heroku.com/articles/deploying-rails-applications-with-the-puma-web-server)
- Scaling Ruby Apps to 1000 Requests per Minute - A Beginner's Guide (https://www.speedshop.co/2015/07/29/scaling-ruby-apps-to-1000-rpm.html)
- Your app server config is wrong @ RailsConf 2017 (https://github.com/speedshop/appserverrailsconf2017)
- Scaling Rails to 125,000 Requests per Minute on Heroku (https://zeemee.engineering/scaling-rails-to-125-000-requests-per-minute-on-heroku-b4128a10a769)

Questions?