O'REILLY®

Velocity

CONFERENCE

BUILD RESILIENT SYSTEMS AT SCALE

Rethinking Quality of Service

Percentages are not People

大胡子 @postwait

CIRCONUS

velocityconf.com

#velocityconf

Quality of Service

- Availability

  the service must be accessible to your users

- Correctness

  the service must perform the function expected

- Performance

  the service must satisfy a user’s productivity goals

Let’s focus on how to measure these things

- Always measure latency in seconds (ms, µs, ns)

  not hours, days or years.

- Always measure throughput in units per second

  if the number is very small, annotate per day or per year.

Measure synthetically

- Perform synthetic measurements (automated use) to measure
  - Correctness
  - Availability

Measure passively

- Passively observe real transactions to measure
  - Performance
  - Availbility

Tactical differences between synthetic & passive measurement

- Synthetic measurements tend to have highly consistent latency at a fixed arrival rate.

- Passive measurements represent reality and include

  - Highly variable rates (no fixed period, yet often Poisson distributed arrival rates)

  - Complex and variable distributions of latency

When you have a lot of data, what question should you ask?

- Assume 10,000 measurements over a minute...
  Should you consider:
  - The average?
  - The variance?
  - The median?
  - Minimum? Maximum?
  - 95th Percentile? 99th? 99.9th? 75th? 25th? 99.5th? …

- Stop… why?

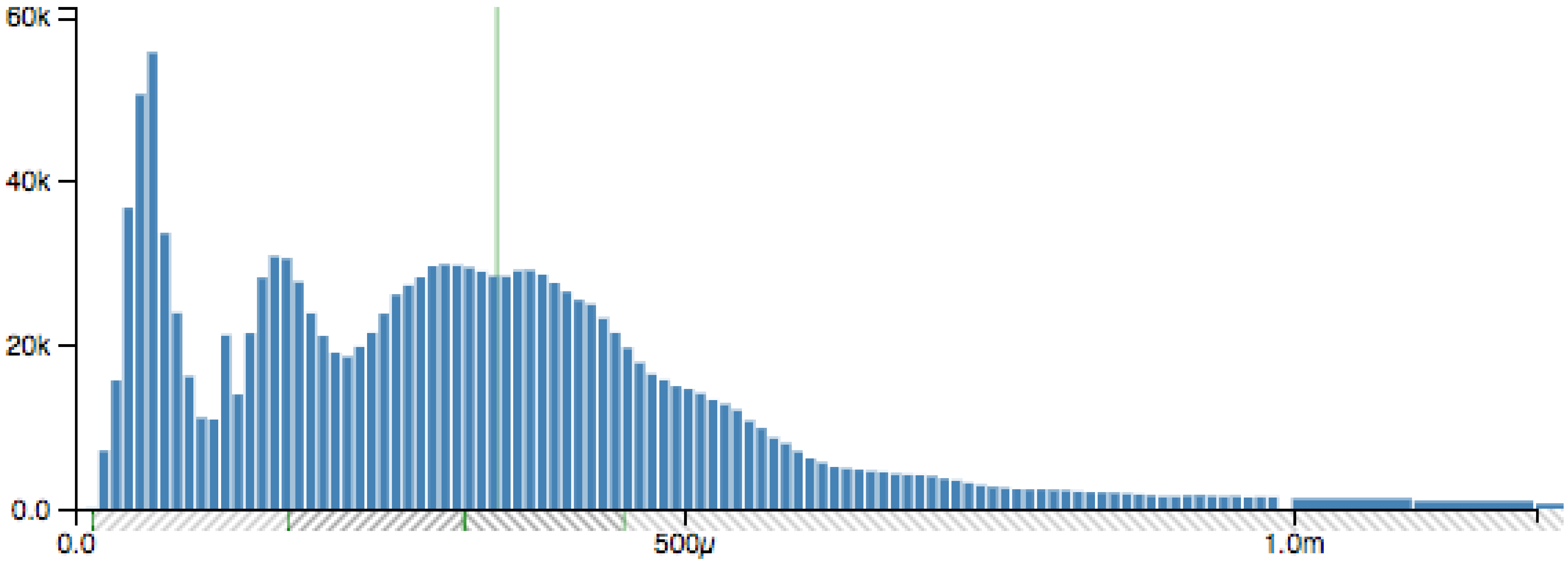
Why do we measure?

- We measure to understand improvement (and degradation)
  - Did we release bad code?
  - Did we fix a latency issue?
  - Are things slower today than yesterday?

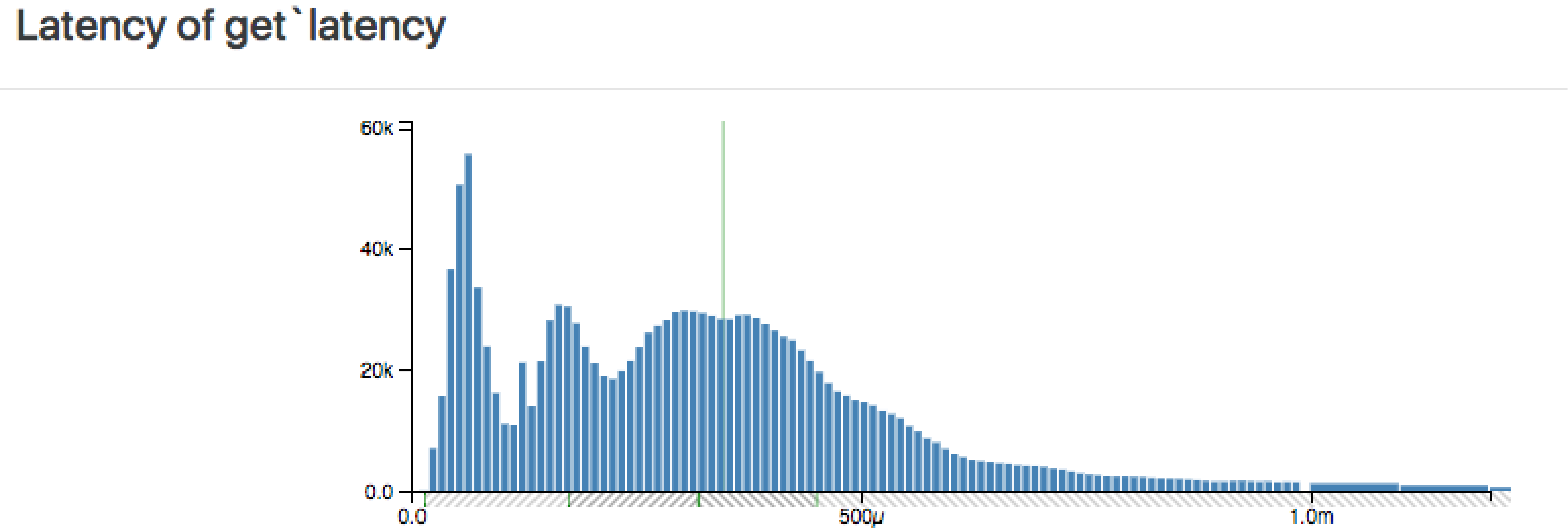
- We measure to discern success
  - Are we fast enough?
  - Are our users happy?

What does observed latency actually look like?

Latency of get`latency

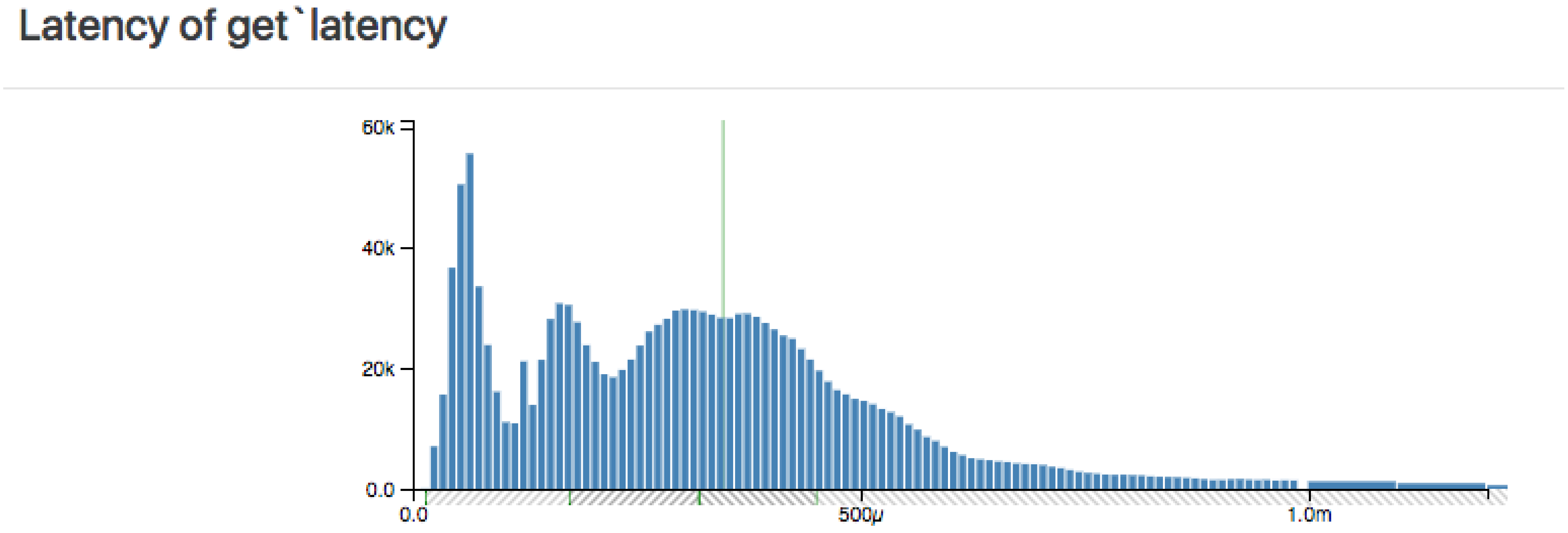


What are all these things?



Latency of get`latency

The “shape” of the histogram indicates a workload



Latency of get`latency

What’s a quantile (or percentile)

- p(99) is the same as q(0.99)
  - Both short for q(SAMPLES, 0.99) as q applies to a set

- Given a set of samples N and a desired quantile Q
- $q(N, Q) \to r$
  - $\geq Q|N|$ samples of N are $< r$ and
  - $\geq (1 - Q)|N|$ samples of N are $> r$
  - any number of samples may be $= r$

We use quantiles

- To describe generalized behavior
- To measure the experience of “most” of our audience.
- To set service level objectives:
  - For N API request latencies, q(N, 0.99) should be less 1ms

The problem

For N API request latencies

q(N, 0.99) should be less 1ms

Is our service level objective

- $q(N,0.99) < 1\text{ms}$

  - The method for selecting N must
    - be consistent and
    - result in a sufficiently sized N
    - (e.g. $N < 100$ would result in some unintuitive results)

  - 0.99 is very different for an N of 10 vs an N of 10,000,000
    you might decide a different quantile is more appropriate later

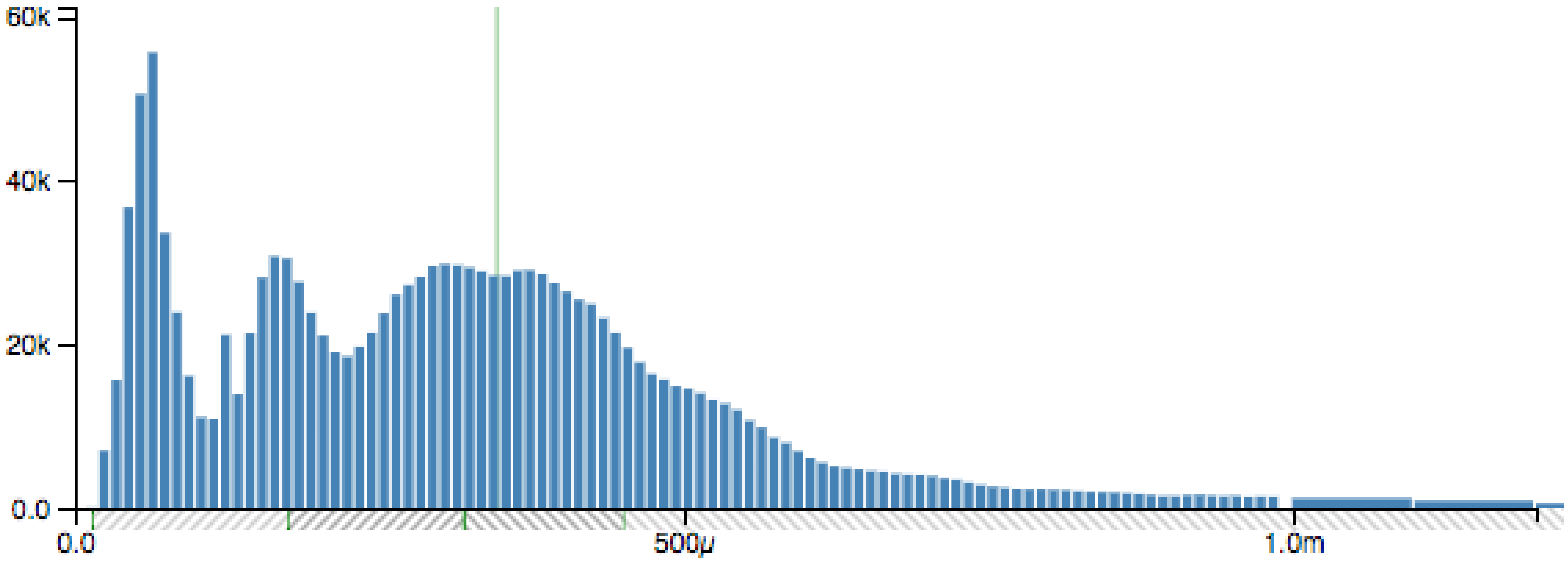
  - 1ms should researched well
    you might decide a different threshold is more appropriate later

Introducing an inverse quantile

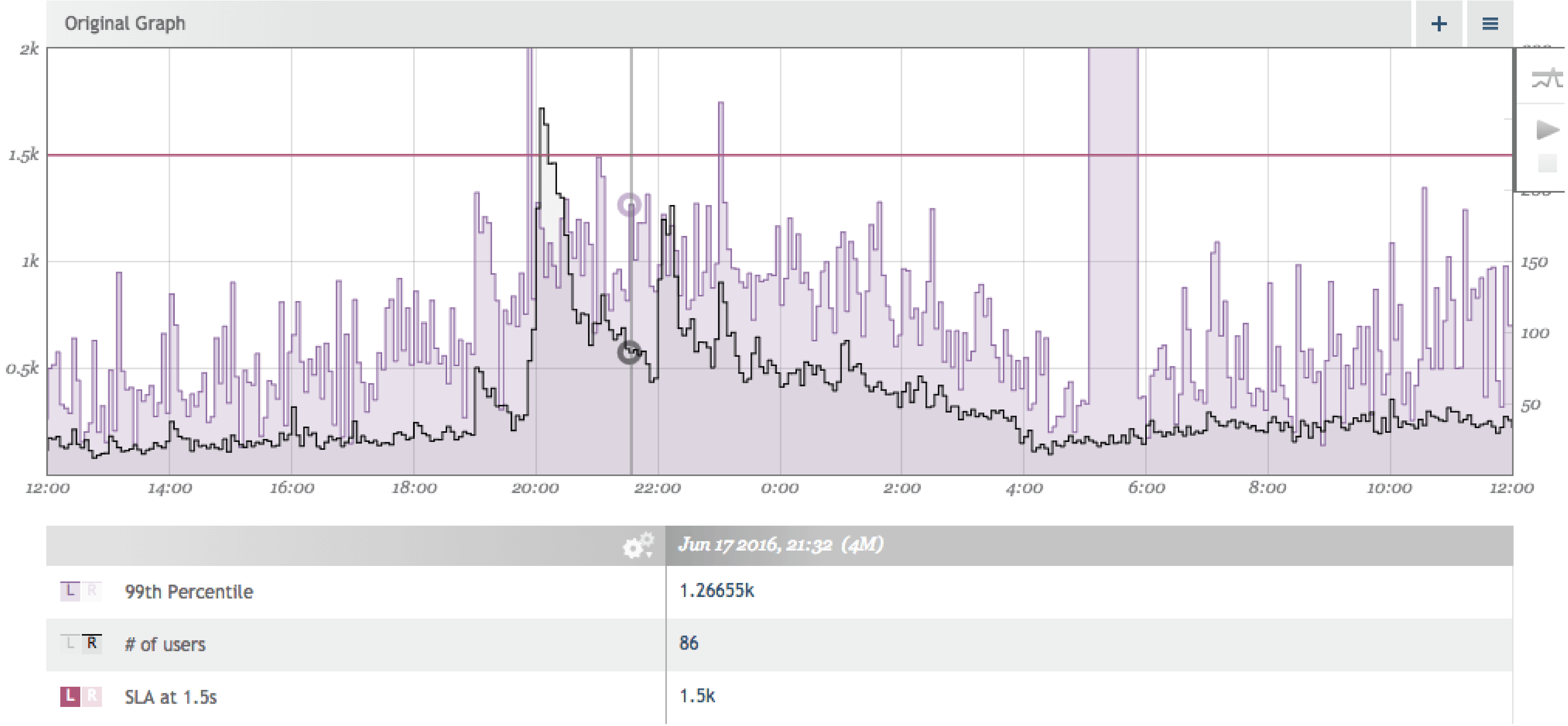
$$q(N,v) = r$$

$$q^{-1}(N,r) = v$$

Latency of get`latency

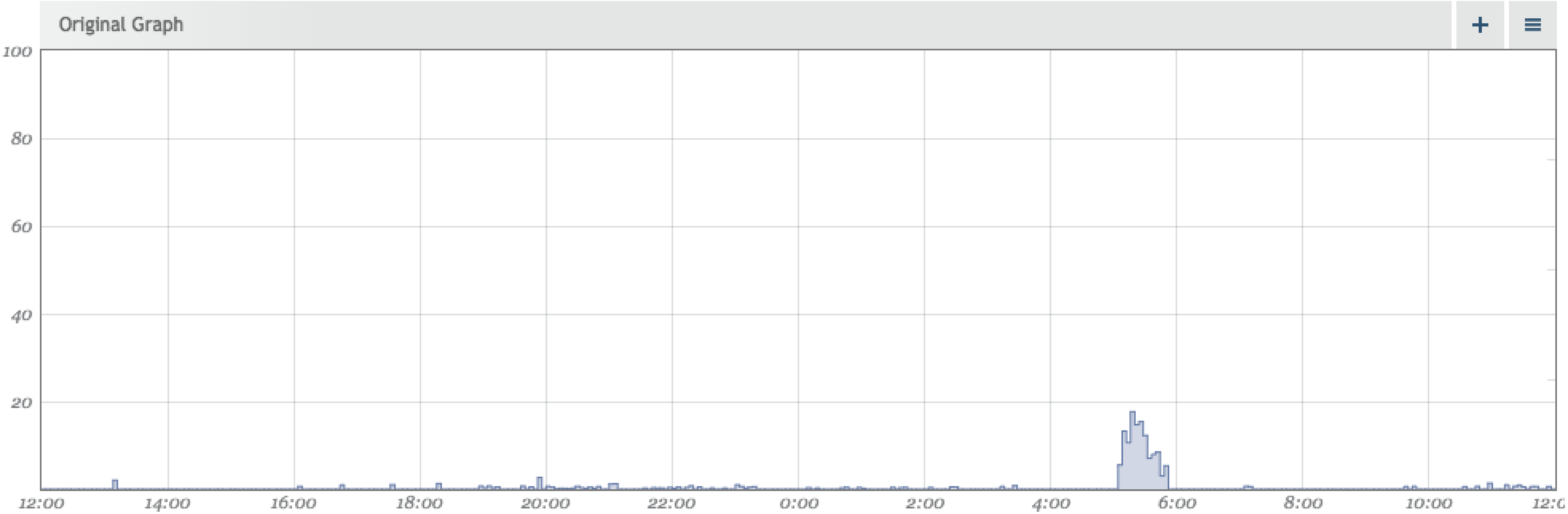


Service Latency q(0.99) vs requests

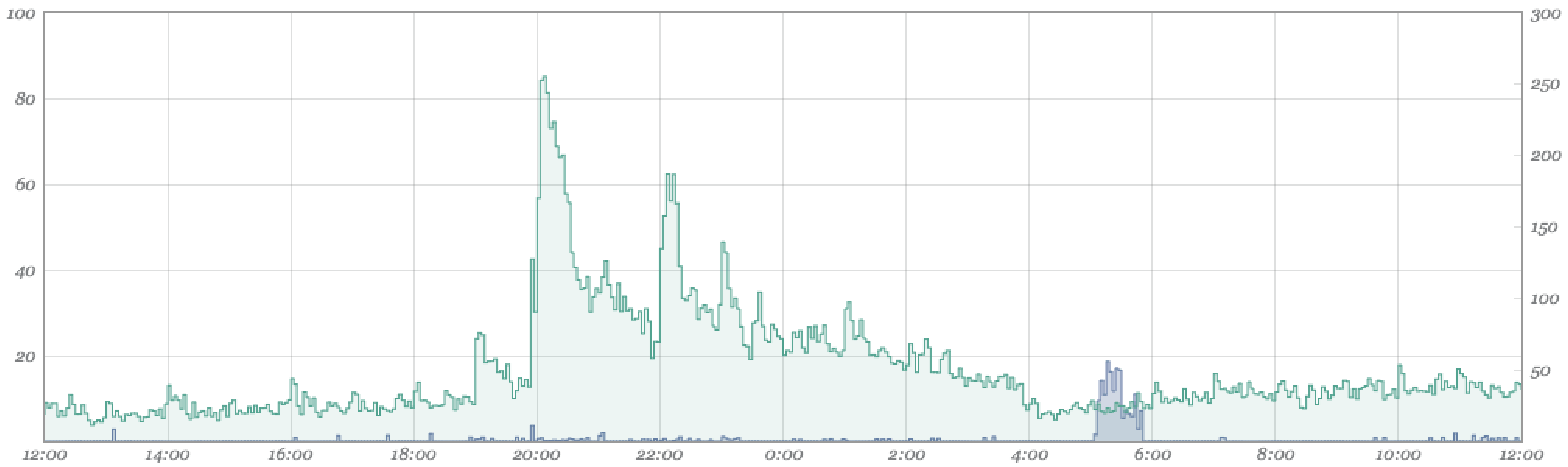


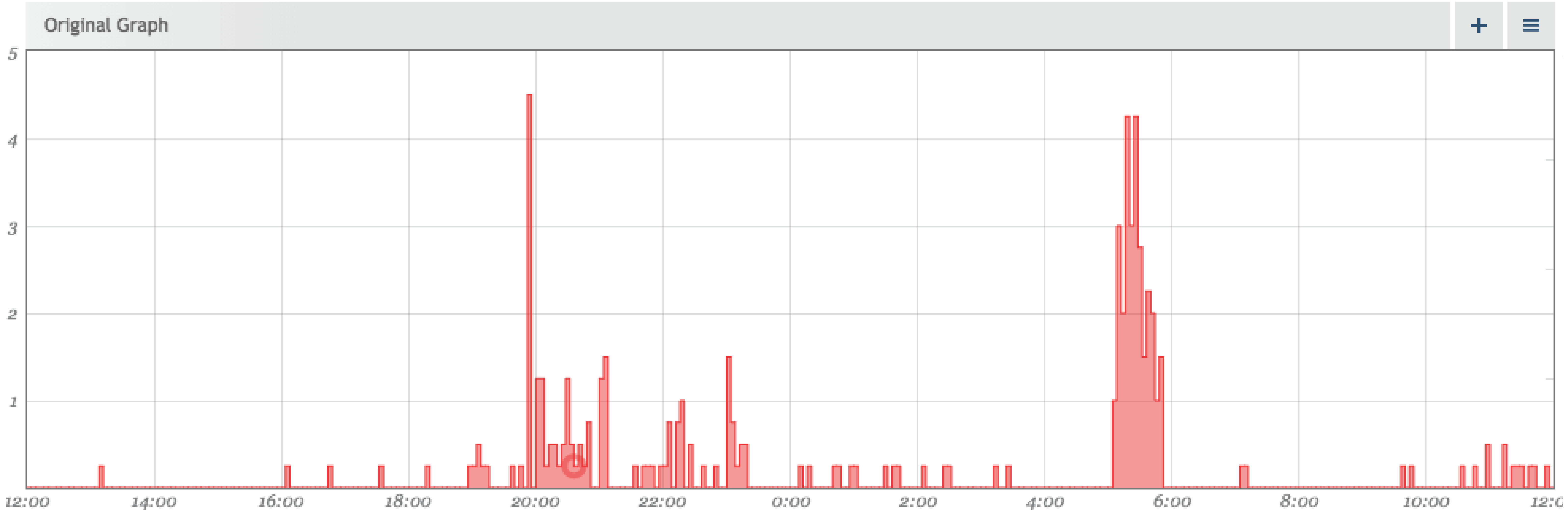
	Jun 17 2016, 21:32 (4M)
L R 99th Percentile	1.26655k
L R # of users	86
L R SLA at 1.5s	1.5k

Percentage of violations: $(1 - q^{-1}(1500ms))*100$

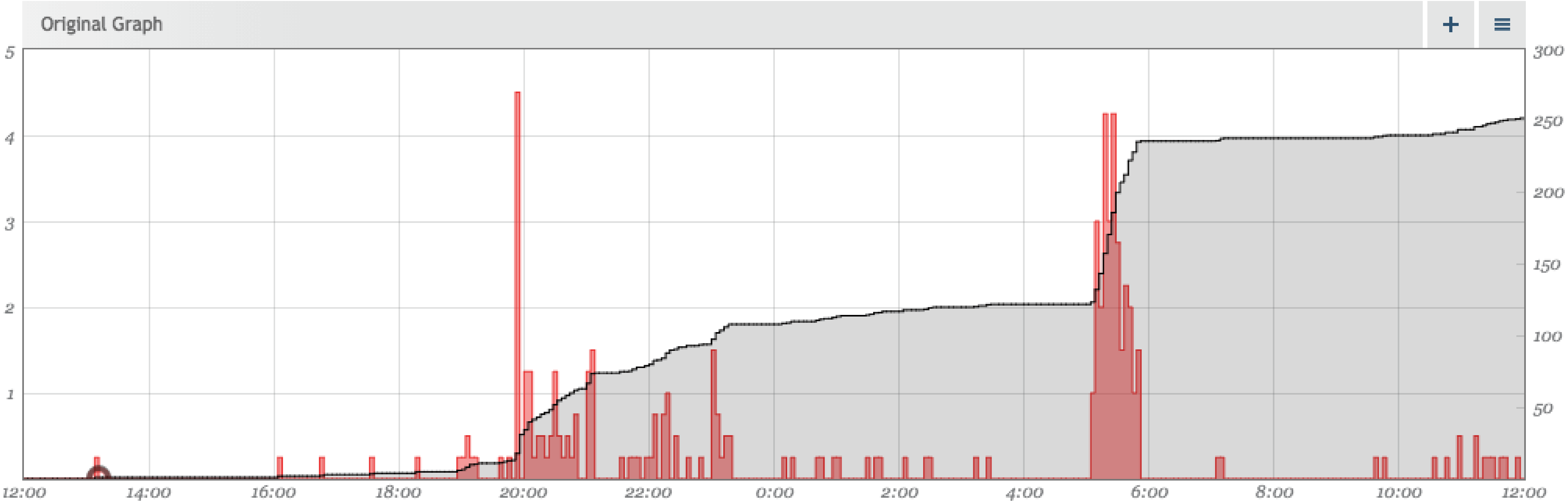


Percentage to actual (pct * requests)





Actual users effected over time (integral)



O'REILLY®

Velocity

CONFERENCE

BUILD RESILIENT SYSTEMS AT SCALE

velocityconf.com

#velocityconf

Thank You

Think about

“how many users have a bad experience”

Instead of

“how bad an experience are a few users having”