CIRCONUS

Latency SLOs Done Right

by @postwait, Founder & CTO @Circonus

CIRCONUS

Service Level Objectives

SREcon 2018 SLO workshop (Google)

Latency SLI

The proportion of valid requests served faster than a threshold

Which requests are valid?
What is the threshold?

CIRCONUS

Service Level Objectives

SREcon 2018 SLO workshop (Google)

Latency SLO

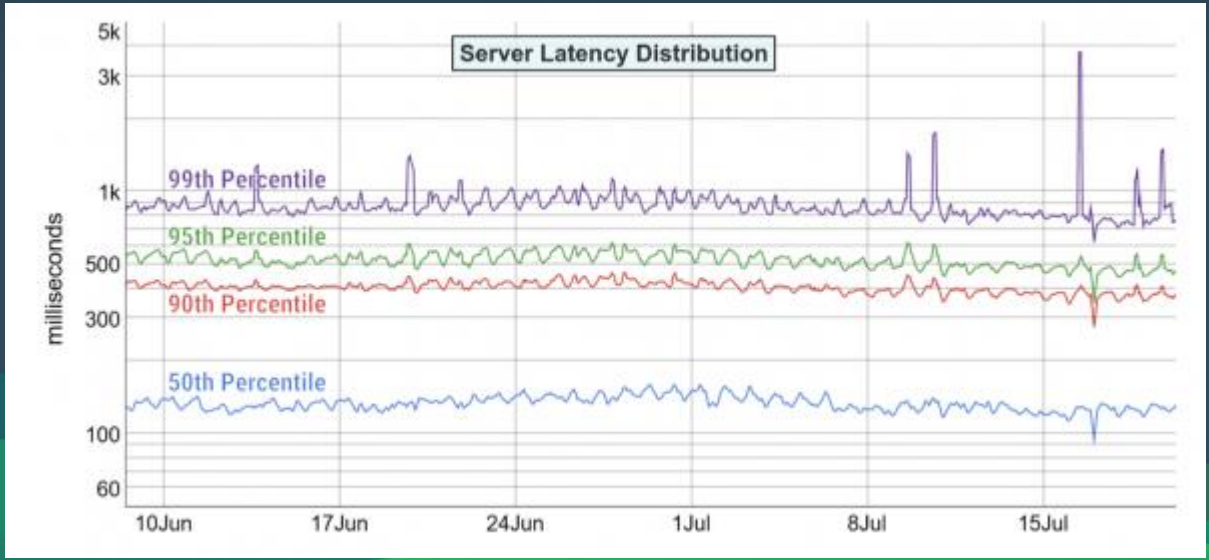
99% of home page requests in the past 28 days served in < 100ms

CIRCONUS

Service Level Objectives

SREcon 2018 SLO workshop (Google)

What is the p90 computed over the full 28 days?

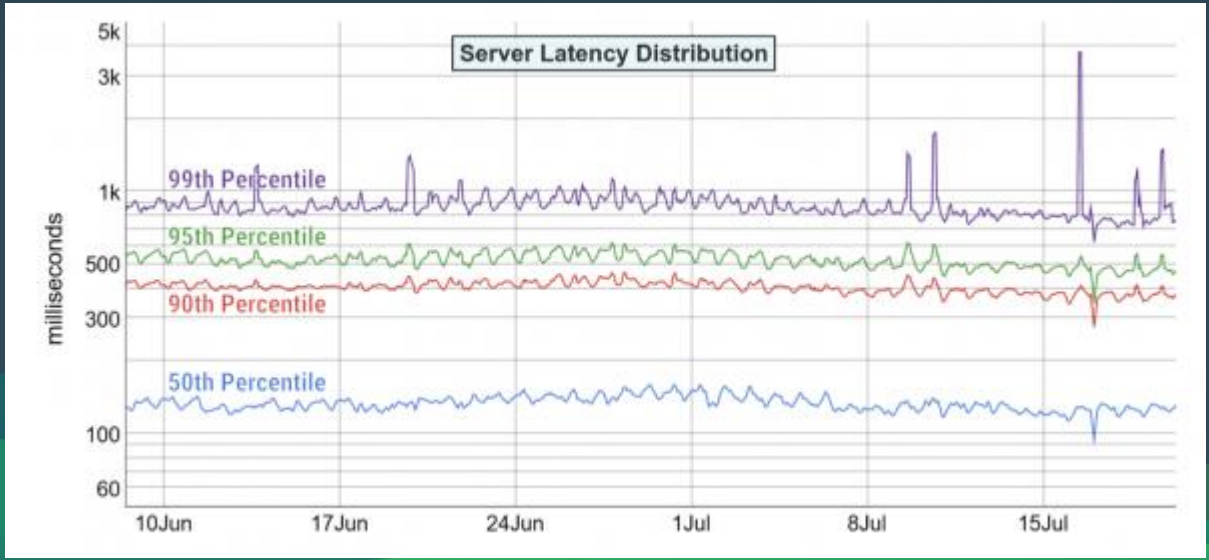


CIRCONUS

Service Level Objectives

SREcon 2018 SLO workshop (Google)

Hint - it's not the average of each p90 sample shown

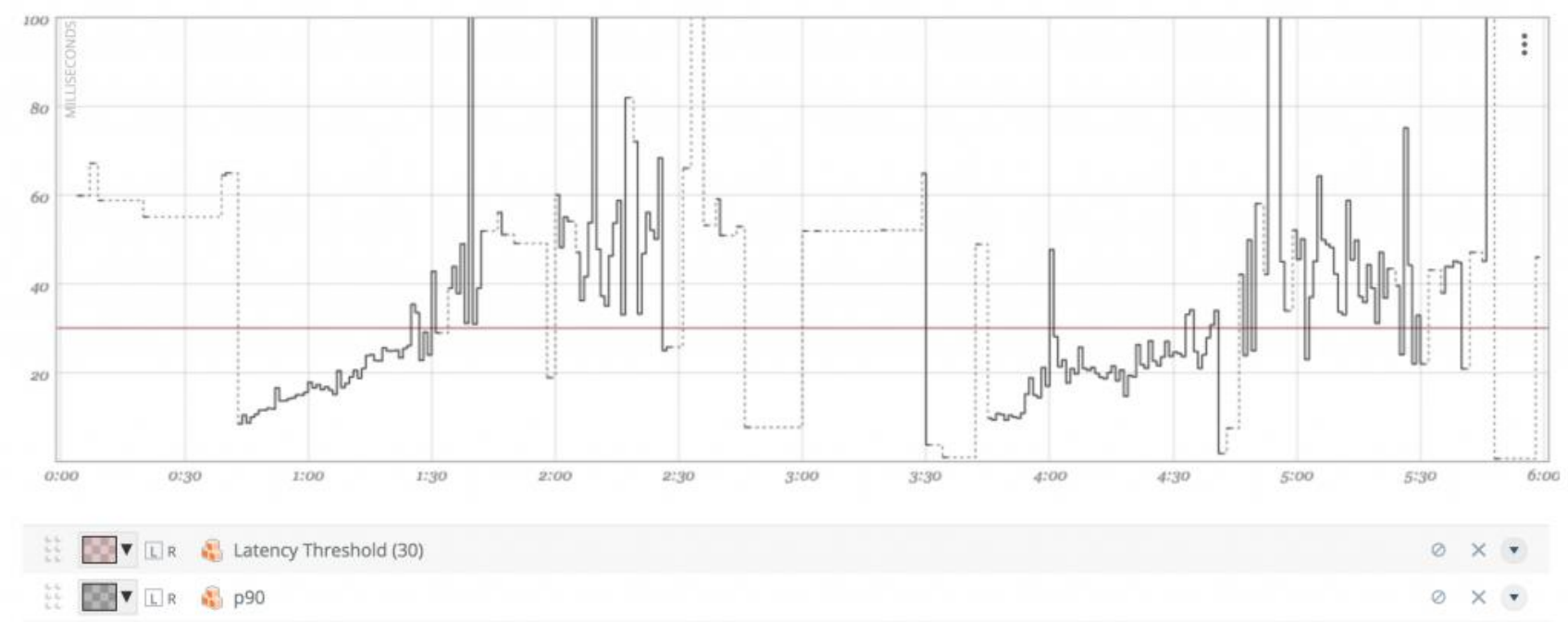


@postwait

CIRCONUS

A more dramatic example

Calculated p90 (10-100ms) != averaged p90 (36ms)



CIRCONUS

Rethinking computing SLO latency

1) Compute the SLO from stored raw data (logs)

2) Count the number of bad requests

3) Use histograms to store latency distribution

CIRCONUS

1) Compute the SLO from stored data

Possible with many tools (Splunk, ELK, awk/grep)

Not always tenable for large volumes of data

Tough to do in real time

CIRCONUS

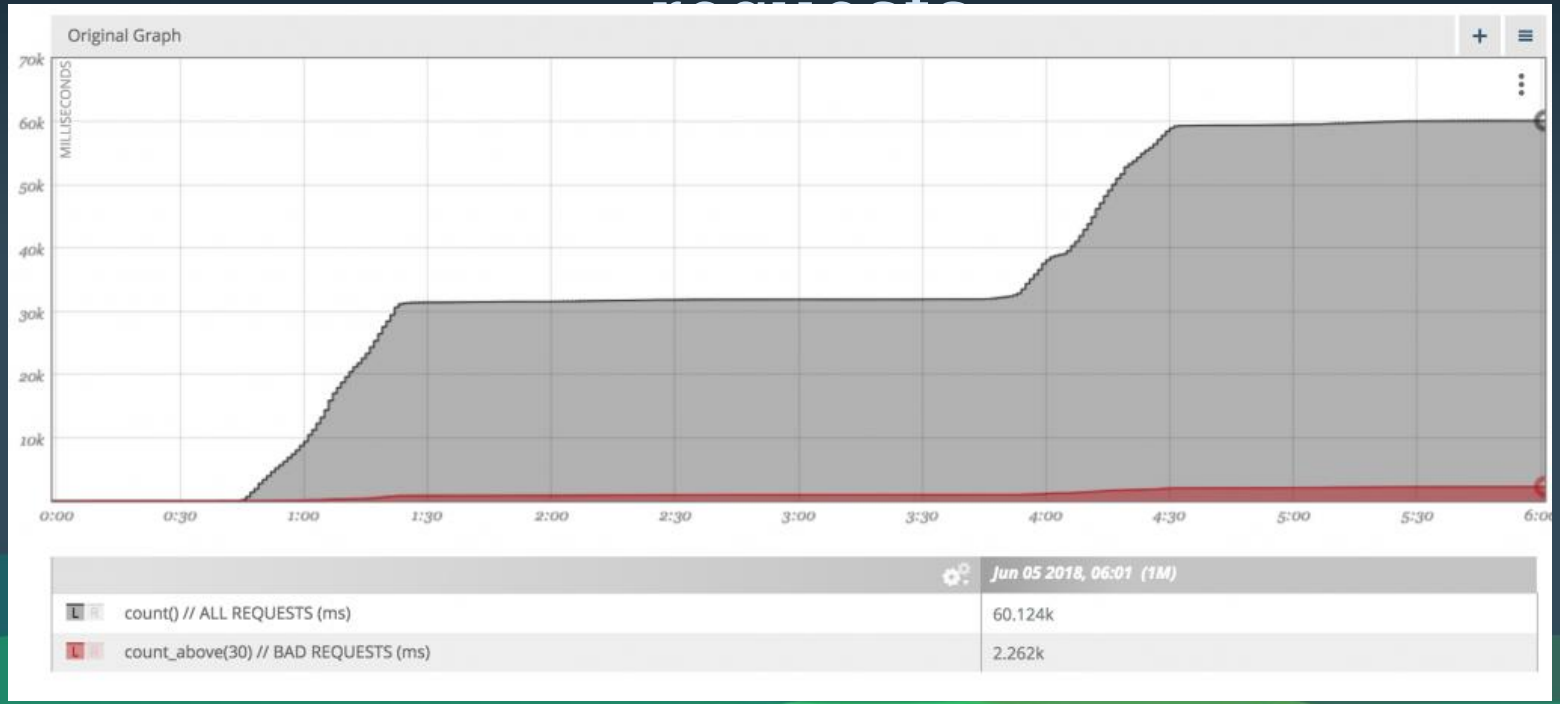
1) Compute the SLO from stored data

An example of calculating percentiles w/ Splunk

```
mydata | stats perc90(responsetime) as response90,
perc99(responsetime) as response99 by
ApplicationName
```

CIRCONUS

2) Count the number of bad requests



CIRCONUS

2) Count the number of bad requests

Percent good = 100 - (2262/60124)*100 = 96.238%

Problem - you have to choose latency threshold up front

If your SLO changes, you can’t analyze historical data

CIRCONUS

3) Using histogram latency data

Histograms can be aggregated across time

Histograms can be used to derive arbitrary percentiles

Bin (bucket) choices should span sample data

CIRCONUS

3) Using histogram latency data

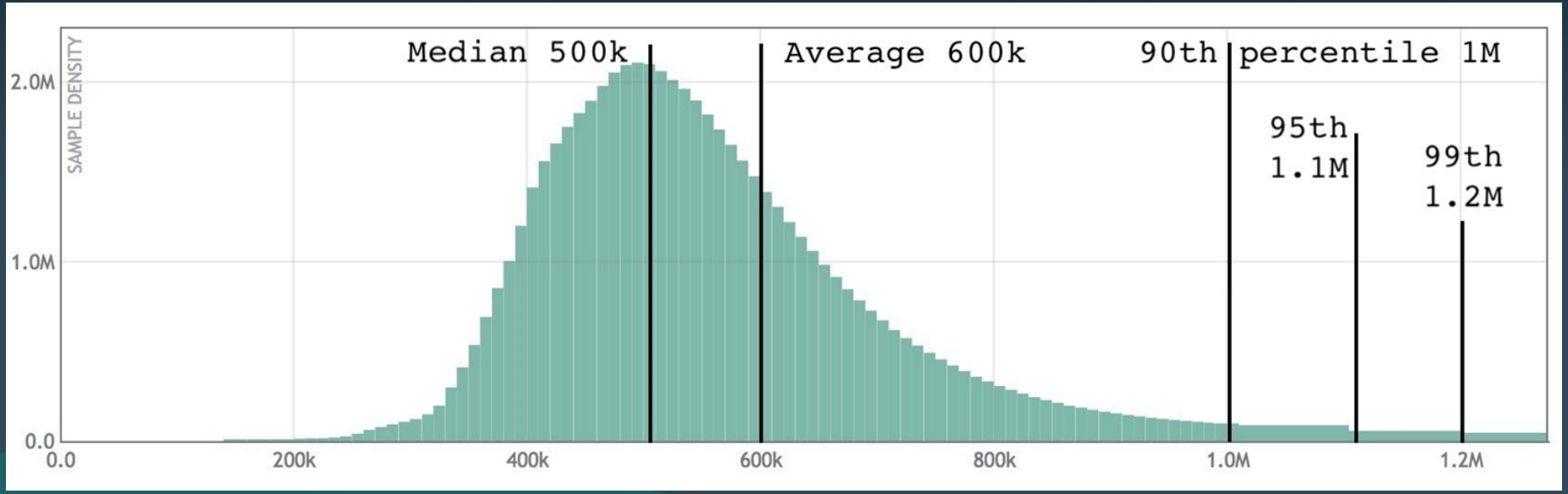
HDR-Histogram – https://HDrhistogram.org

Circllhist – https://github.com/circonus-labs/libcircllhist/

t-digest – https://github.com/tdunning/t-digest

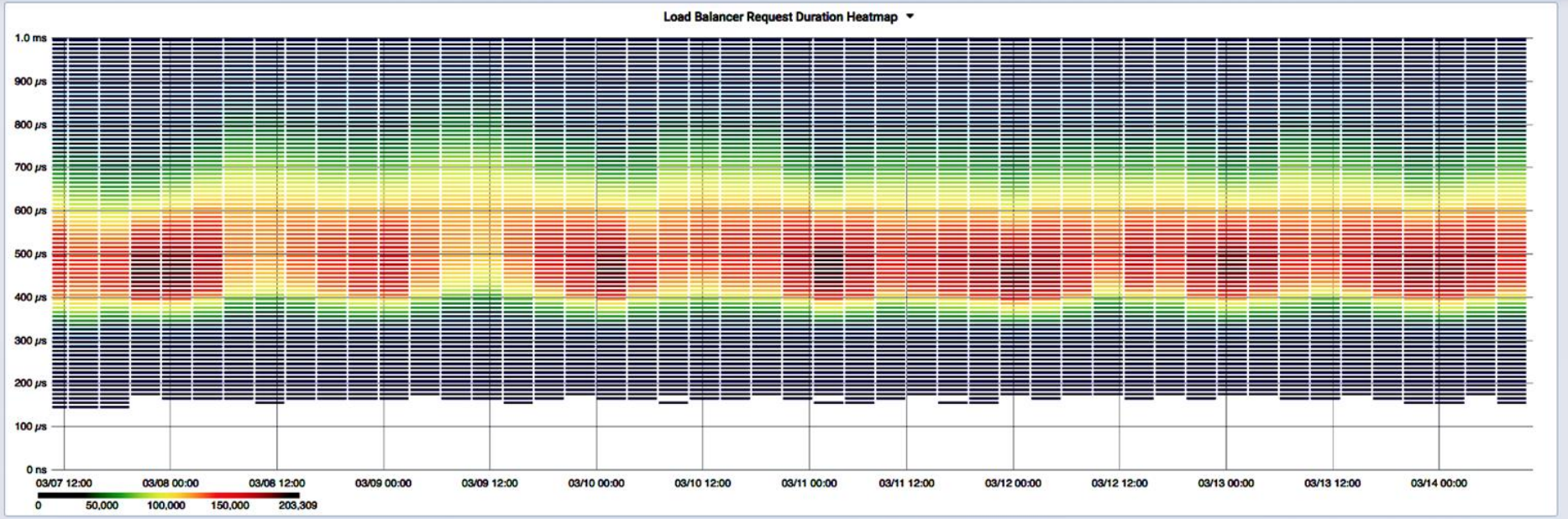
CIRCONUS

3) Using histogram latency data



CIRCONUS

3) Using histogram latency data



CIRCONUS

3) Using histogram latency data

99% of home page requests in the past 28 days served in < 100ms

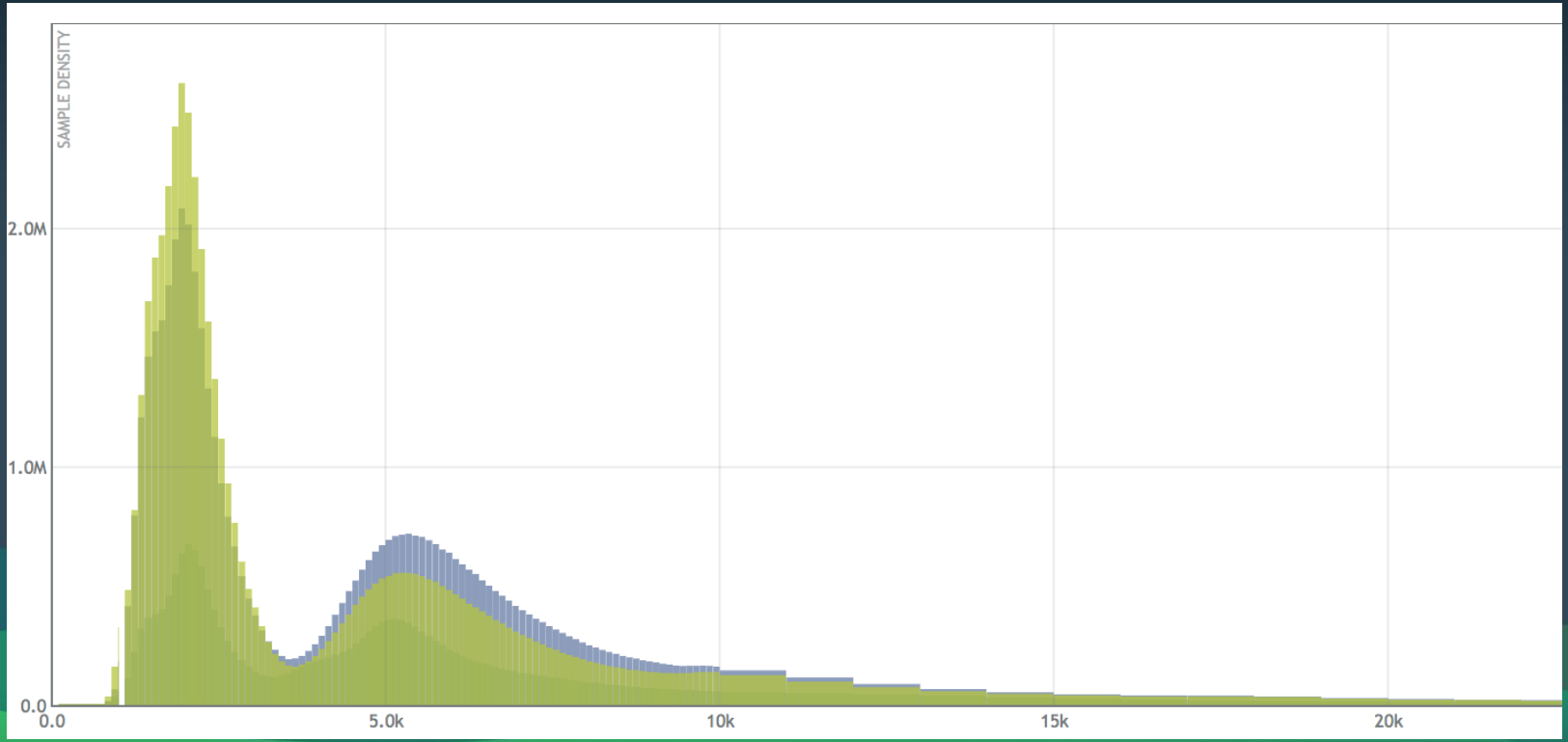
% requests = count_below(100ms) / total_count * 100

ex: 99.484 percent faster than 100ms

This is an **inverse percentile**.

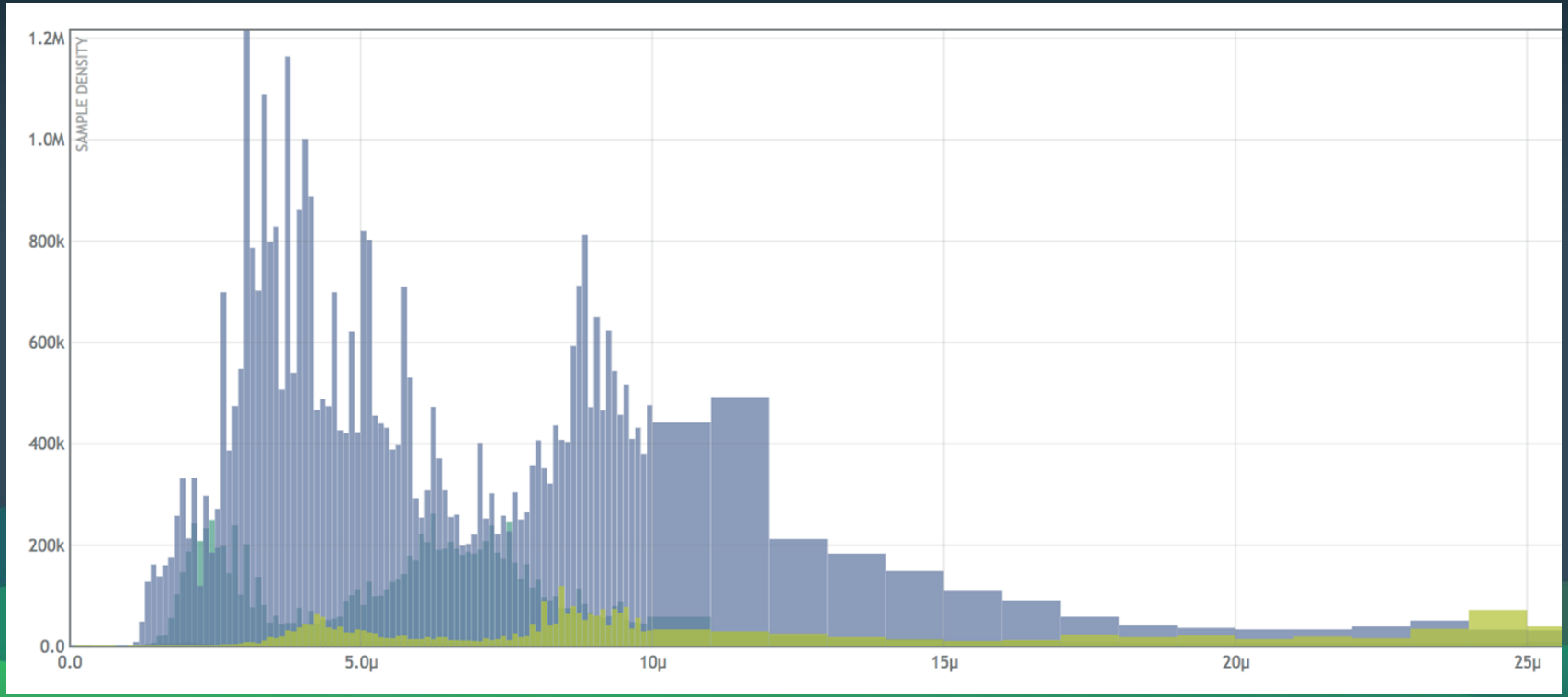
CIRCONUS

3) Using histogram latency data



CIRCONUS

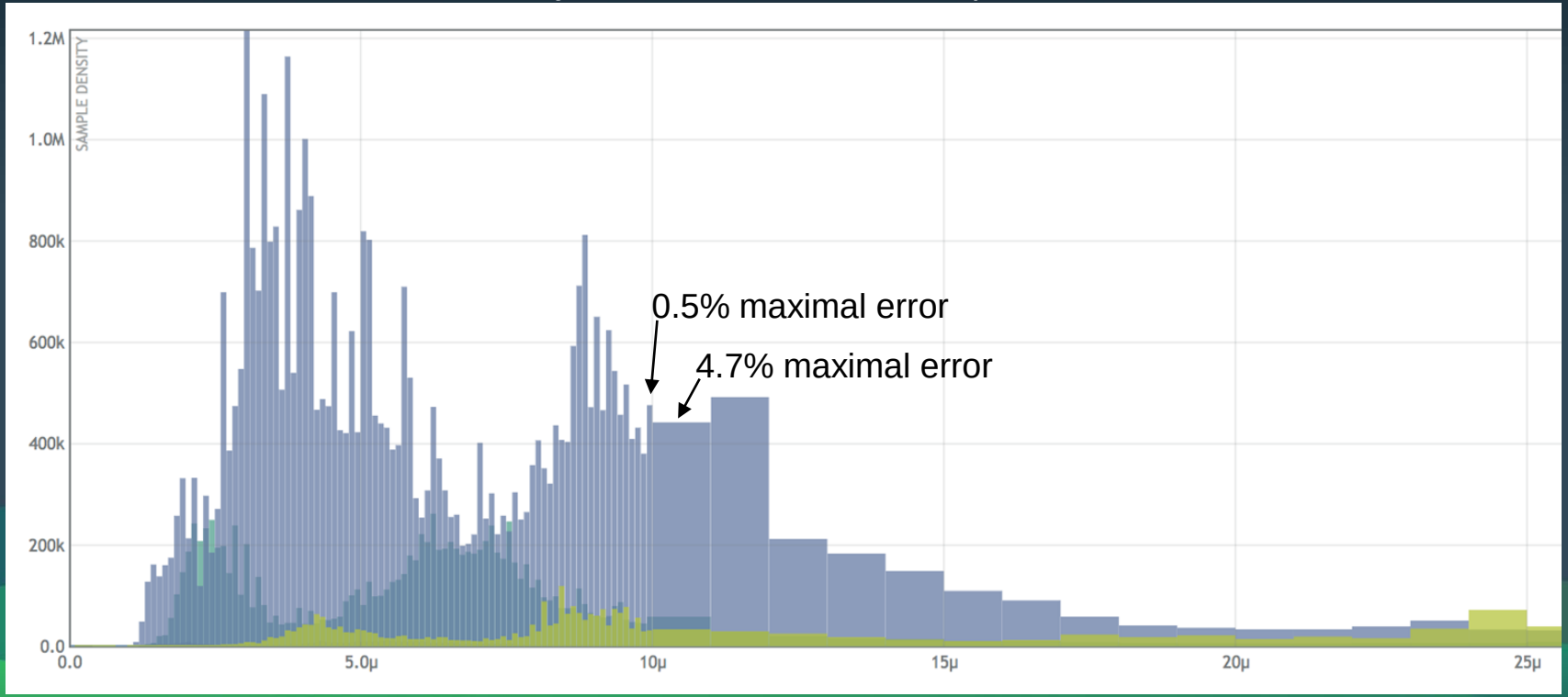
3) Using histogram latency data



CIRCONUS

libcircllhist: C, Java, Go, Javascript

Envoy uses libcircllhist, for example.



CIRCONUS

Framing SLOs as quantiles is backwards

- When we say:
  99.5%
  of requests should be faster than
  100ms

- We don't care as much about how fast the 99.5^{th}% is... p(99.5) or q(0.995)

- We actually care what percentile is at 100ms... $q^{-1}(0.1)$

CIRCONUS

Framing SLOs correctly is important

- If you are “doing SLOs”
  (and budgets around them)
- You are literally investing time, money, and focus based on the answers to math questions.
- Ask the right questions.
- Do the math right.
- Histogram representation is the “right” statistical representation for these questions.
- t-digest and moment sketches are beautiful and awesome and powerful, but they help answer ***different*** questions… and answer these questions poorly.

CIRCONUS

Framing SLOs is iterative

- Since we're literally investing around these numbers...

  Why is 99.5% at 100ms right?

  And not 99.2% at 115ms?

- If you can't answer this question...

  maybe you shouldn't take your SLO so seriously.

- By keeping historical data with the right granularity to answer these "new" proposals for SLOs... you can iteratively optimize your parameters.

CIRCONUS

SLOs... the undiscussed problem

- There are two times that are important.

  1) the period over which you calculate your quantile.

  2) the period over which you calculate your objective success.

- SLOs don't look like:
  99.5% under 100ms is an incomplete phrasing.

- They actually look like:
  99.5% under 100ms over any five minute period... and
  99.9% of those are satisfied in a rolling 28 day period.

(yes, that's now 4 parameters to select correctly)

CIRCONUS

Tool Choices

- You need correct math
- Use a tool that either uses raw data or binned histograms.
- You should be able to quantify statistical error in every answer you get back.
- You need history
- You SLOs (at first) will be offensively arbitrary. In order to improve them you must be able to re-analyze your data.
- You need correlation
- You can’t refit parameters without the data your attempting to fit to.

CIRCONUS

Questions?

Tweet me @postwait

Ask us anything on Slack at http://slack.s.circonus.com/