Google

History-Based Latency Prober Tuning

Jeff Borwey (jborwey@google.com)
March 15, 2022

sre.google • twitter.com/googlesre

Site Reliability Engineering

Monitoring 101: Probers!

Jeff Borwey
SREcon22 Americas
March 15, 2022

What are Probers

- Fixed workload
- Executed with a regular cadence
- Without knowledge of internals

Prober Design Decisions

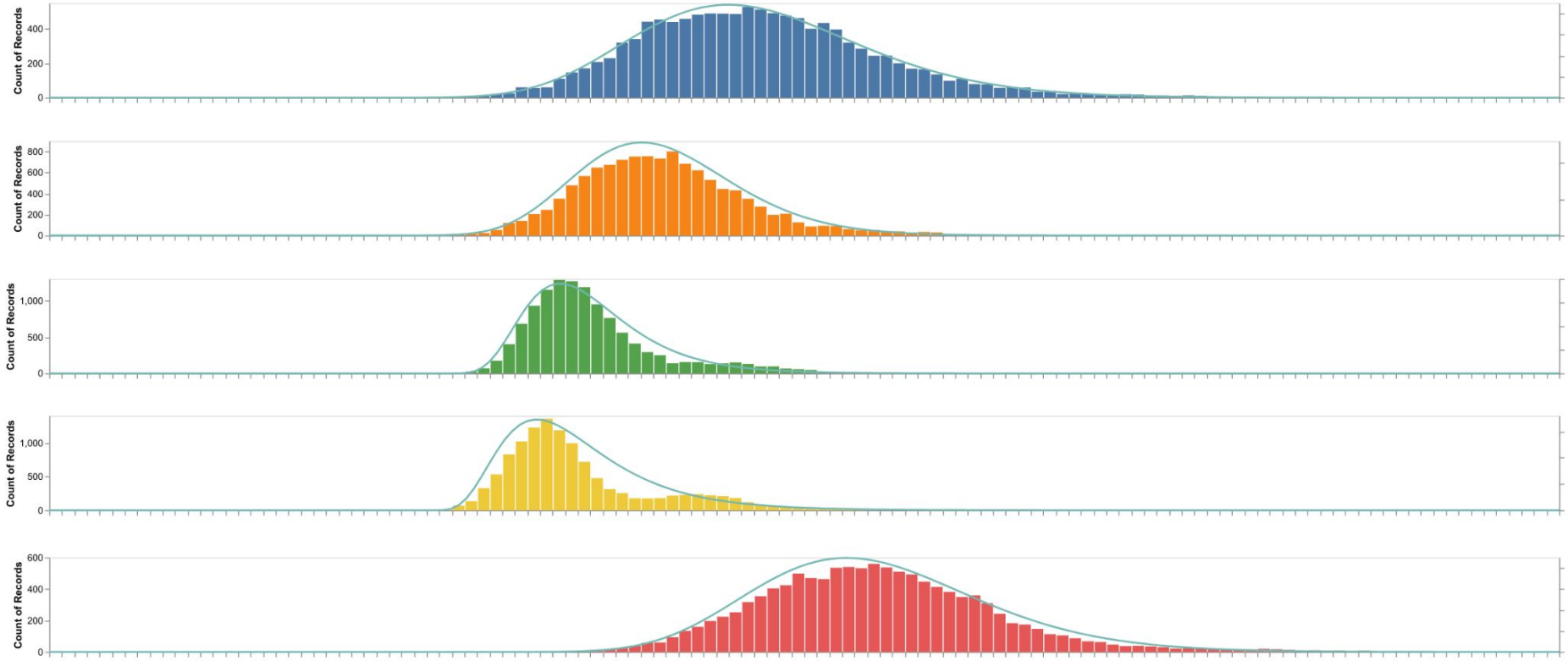
Key Design Decisions

1. What code paths should be exercised?
2. How frequently should it be run?
    a. Cost/Detection Latency trade off
3. **How do you alert on this signal?**
    a. **What is an acceptable latency?**
        i. Does this change over time?
        ii. Does this change per environment?
    b. **How do we control false-positive rate?**

How is it done in practice?

1. Identify an area of the system with low visibility
    a. New feature
    b. Reoccuring source of regressions/outages
2. Craft a specific workload that exercises the relevant code paths
3. Throw it into your framework of choice
4. Use “domain expertise” to pick a threshold
5. Get paged too much…
6. Increase the threshold

Now scale that to 100’s of environments


Count of Records
400
200
0
Count of Records
800
600
400
200
0
Count of Records
1,000
500
0
Count of Records
1,000
500
0
Count of Records
600
400
200
0

Better Probing Through History!

Jeff Borwey
SREcon22 Americas
March 15, 2022

A Better Solution: Look to History!

Alerting thresholds via **outlier detection**:

- Model historical performance
- Derive threshold for Out-Of-Distribution samples
- Adjust windowing
- Backtest!

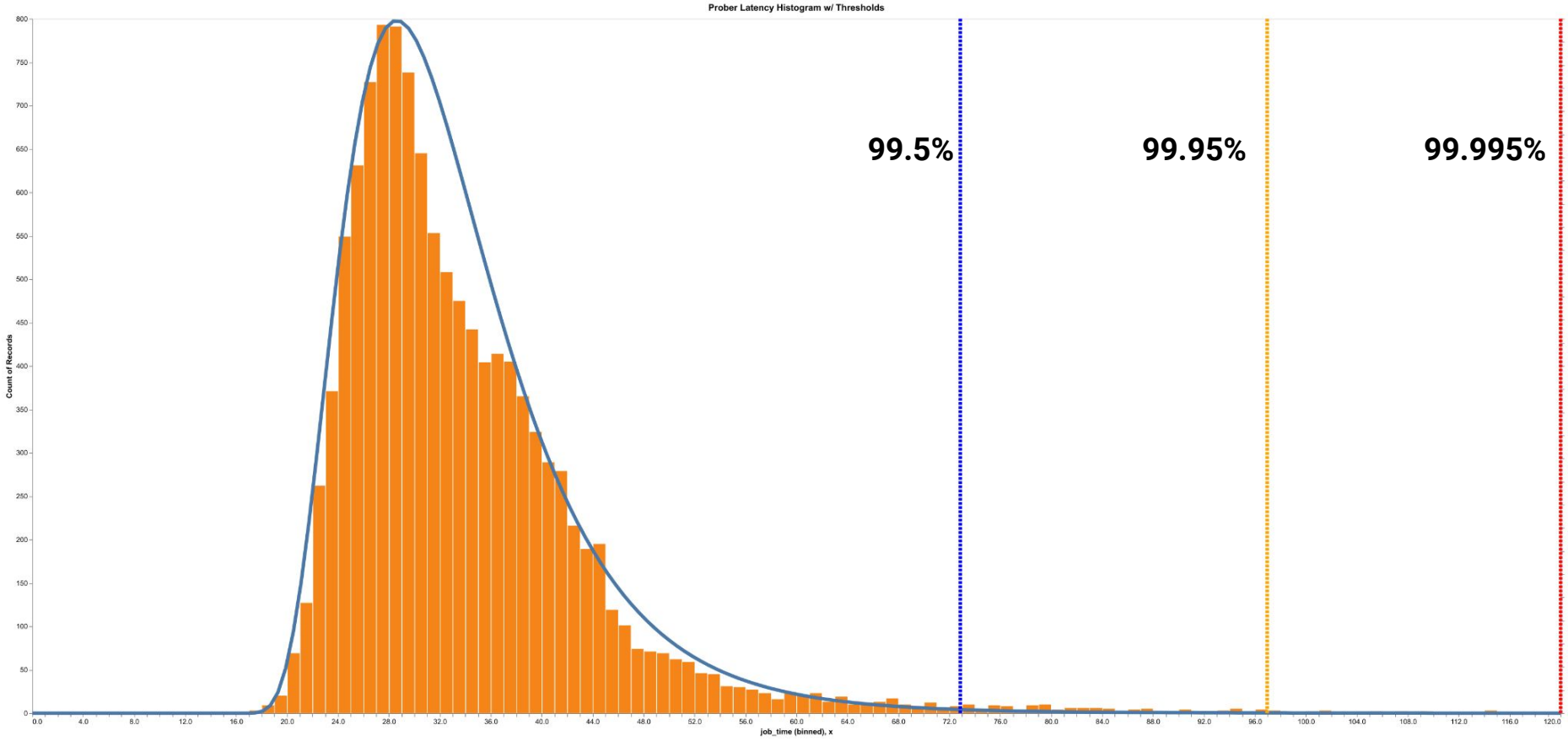
Parametric approach

- Model latency distribution per environment
- Pick a high percentile of the distribution (99.5%, 99.95%, etc)
- Log-normal often works well for latencies
- Requires trimming outliers before fitting

$$log(X - l) \sim N(\mu, \sigma^2)$$

$$X \sim Lognormal(\mu, \sigma^2, l)$$

μ is the mean, **σ^2** is the variance, **l** is the location offset


Prober Latency Histogram w/ Thresholds
99.5%
99.95%
99.995%
Count of Records
job_time (binned), x

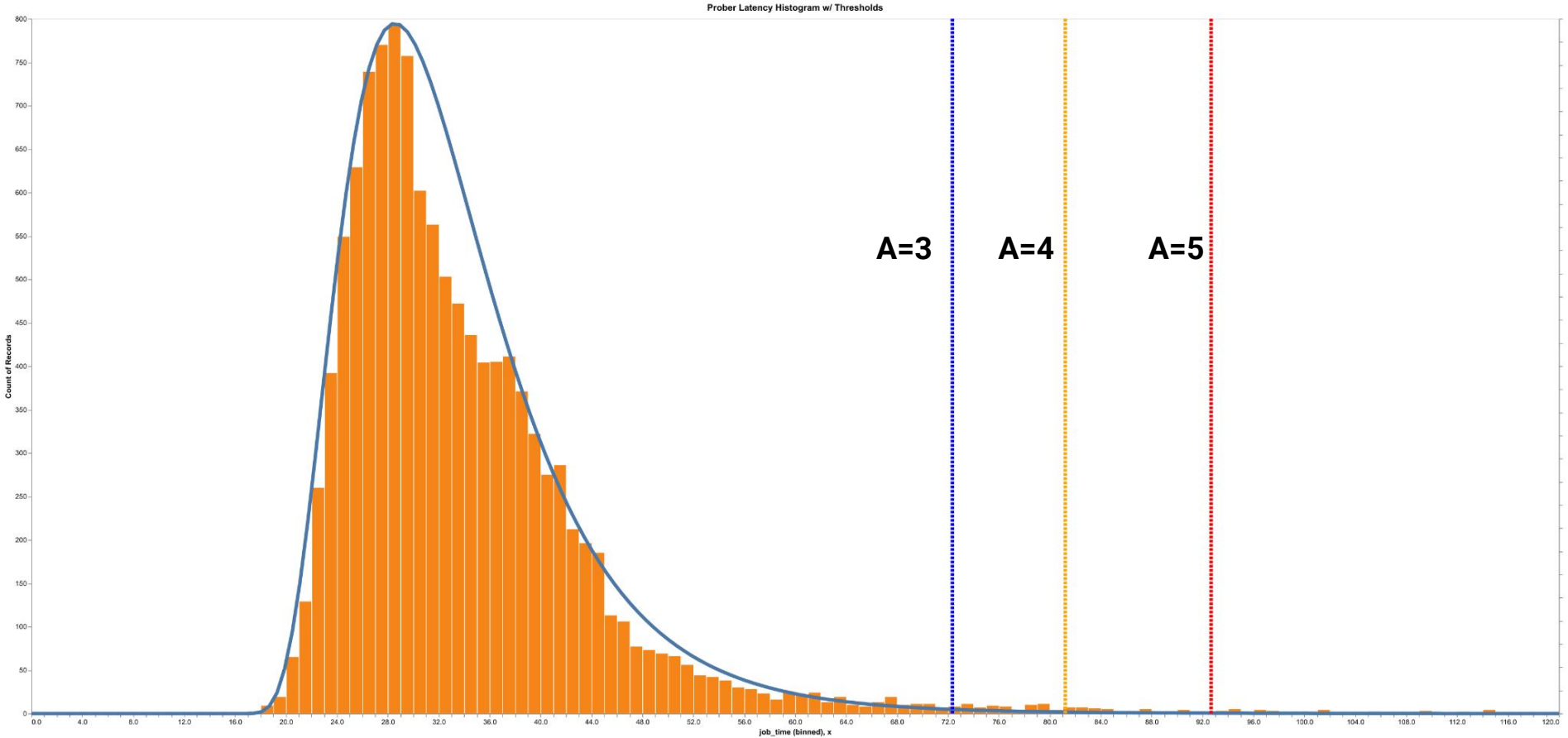
Non-parametric approach

- Compute **Interquartile Range (IQR) (75%-25%)**
- Compute **skewness** using **medcouple (MC)**
- Skew-adjusted box-plot [1]

$$threshold = Q_3 + 1.5 * e^{A*MC} * IQR$$

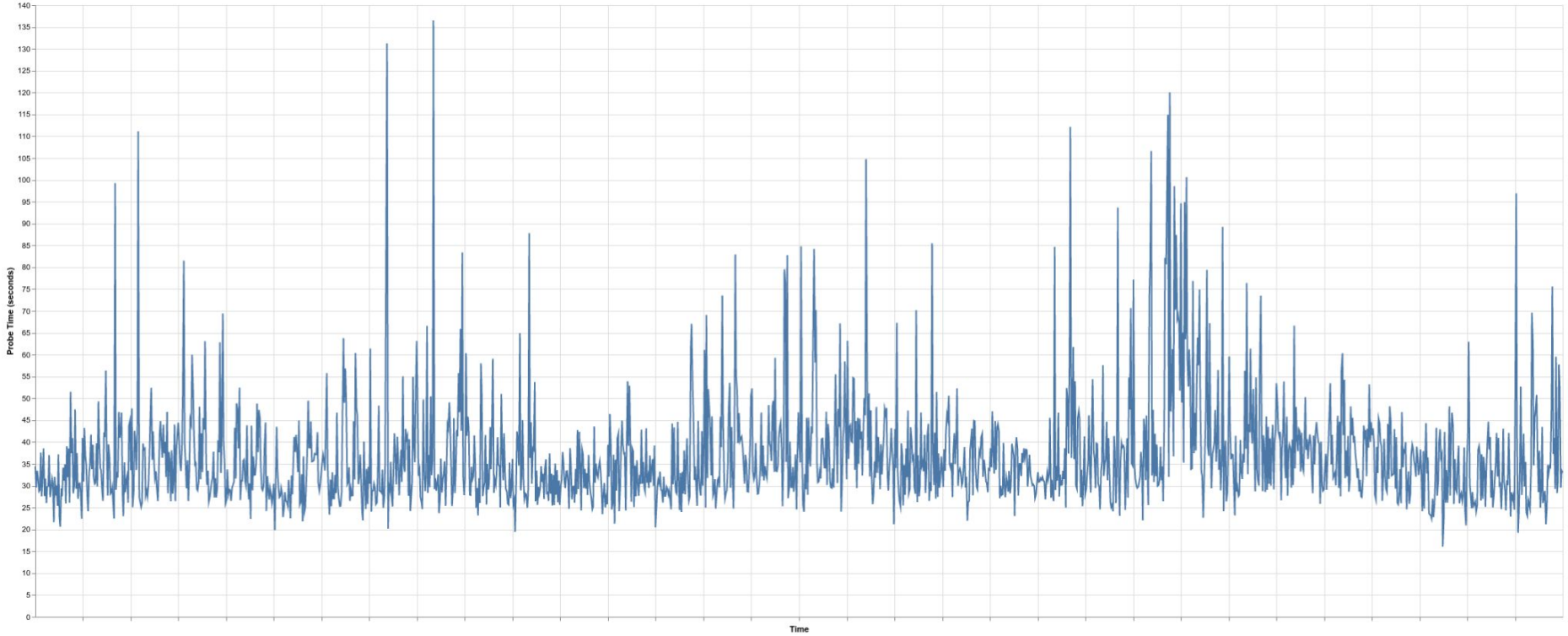
Q_3 is the 75%, **A** is a tuneable parameter

[1] Hubert, M.; Vandervieren, E. (2008). “An adjusted boxplot for skewed distribution”. *Computational Statistics and Data Analysis*. **52** (12): 5186–5201.

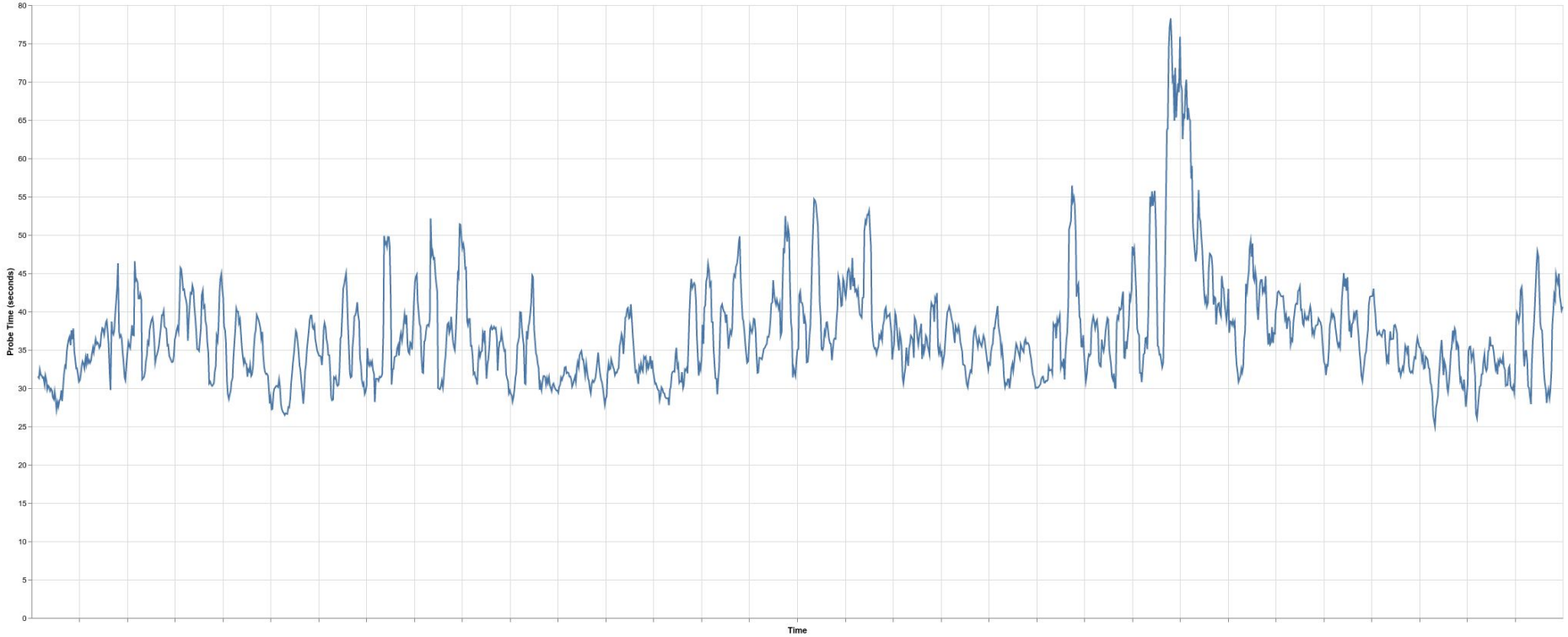

Prober Latency Histogram w/ Thresholds
A=3
A=4
A=5
Count of Records
job_time (binned), x

From Samples To Alerts

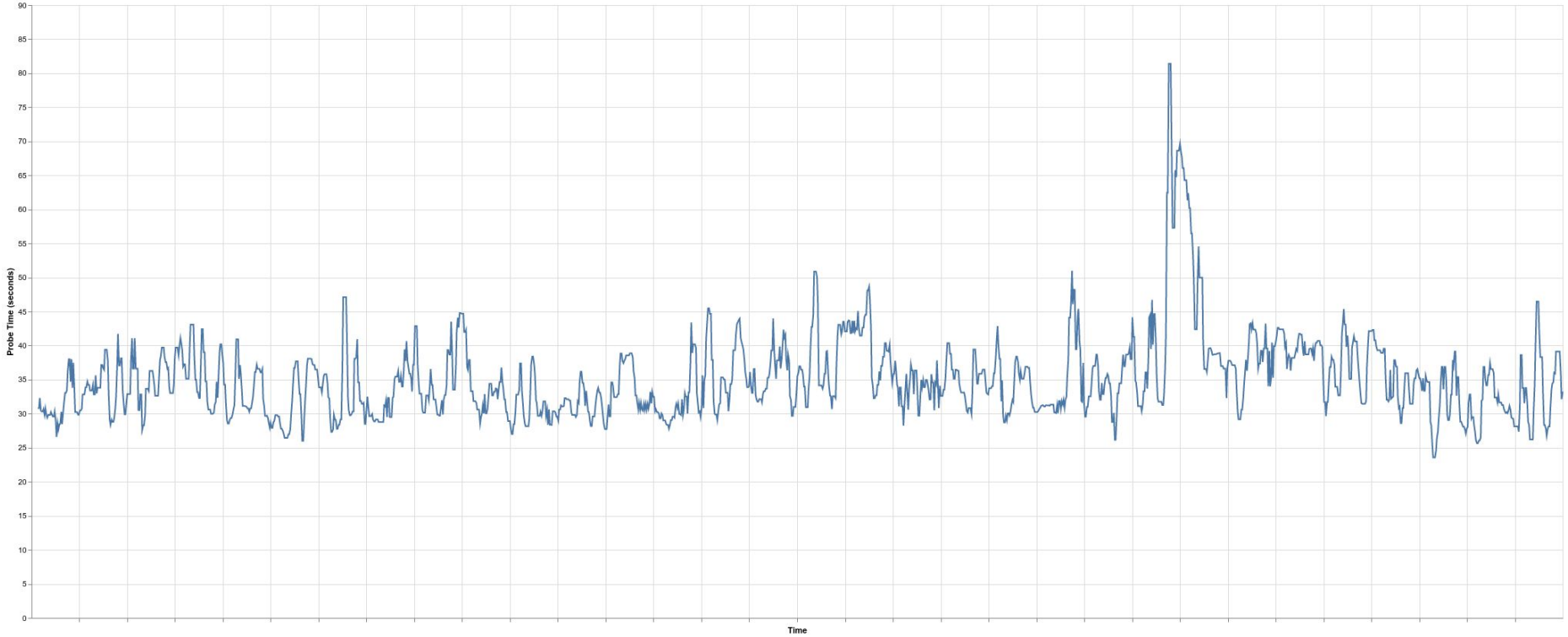
Windowing

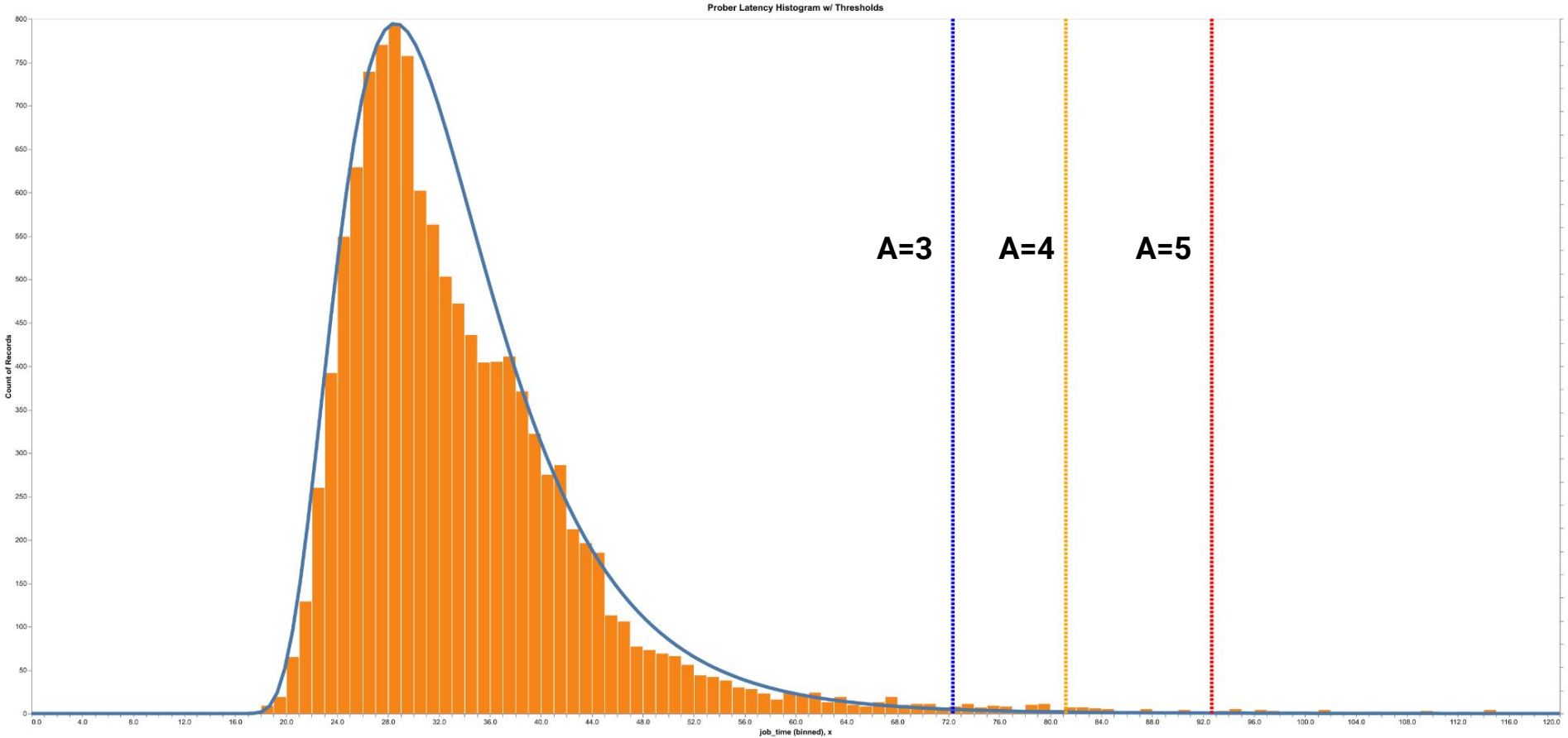

Probe Time (seconds)
140
135
130
125
120
115
110
105
100
95
90
85
80
75
70
65
60
55
50
45
40
35
30
25
20
15
10
5
0
Time

Windowing (Average)

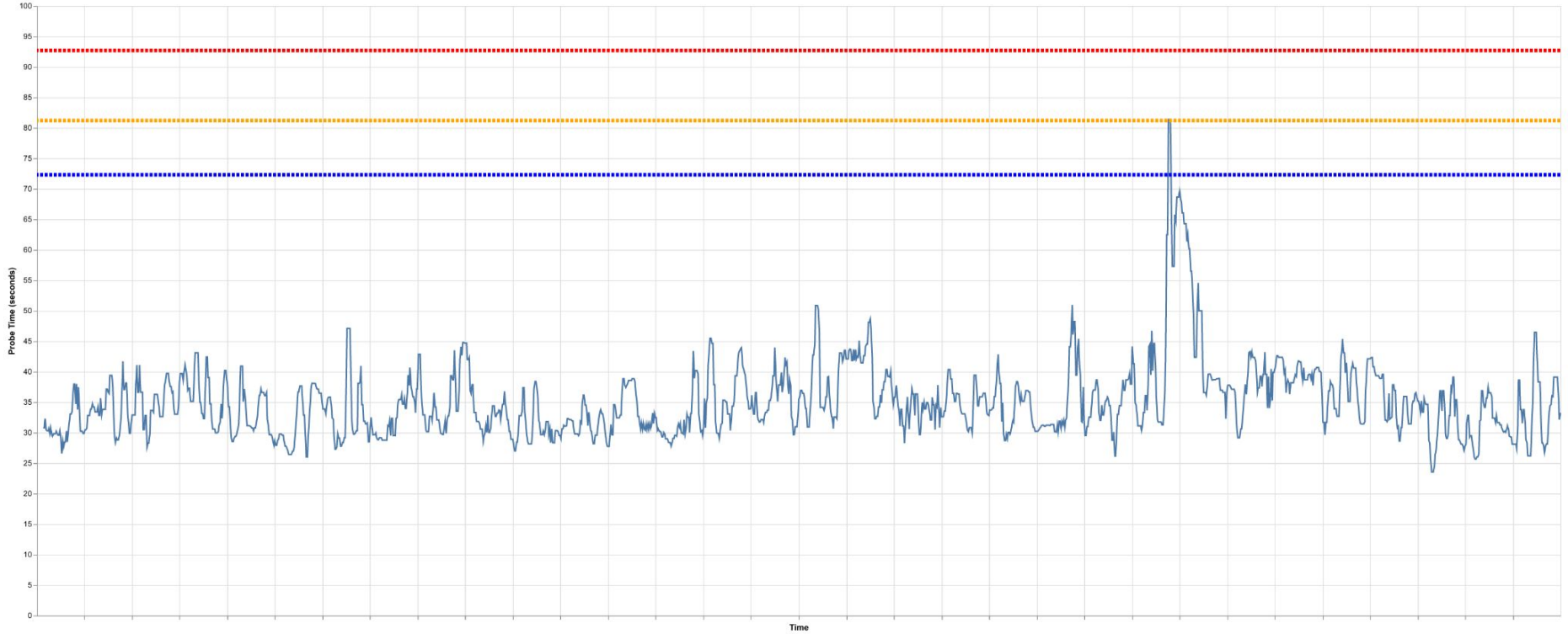


Windowing (Median)

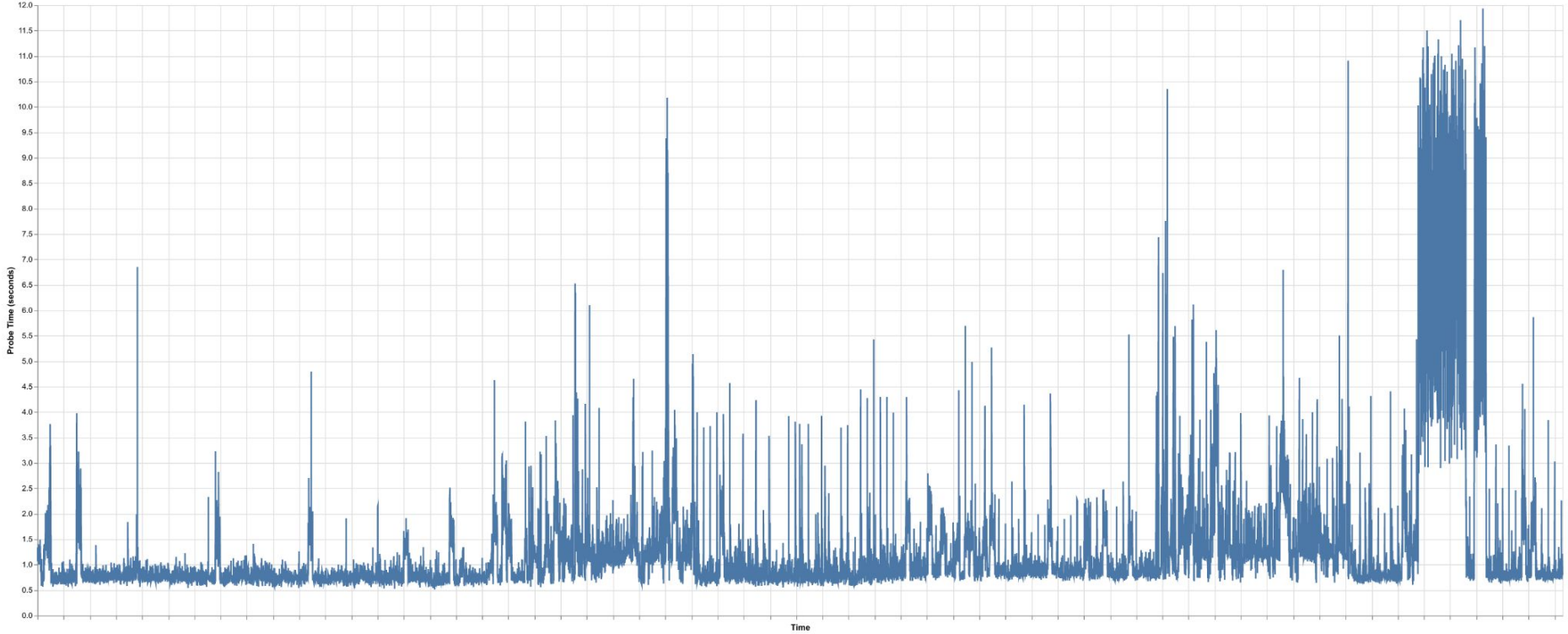



Prober Latency Histogram w/ Thresholds
A=3
A=4
A=5
Count of Records
job_time (binned), x

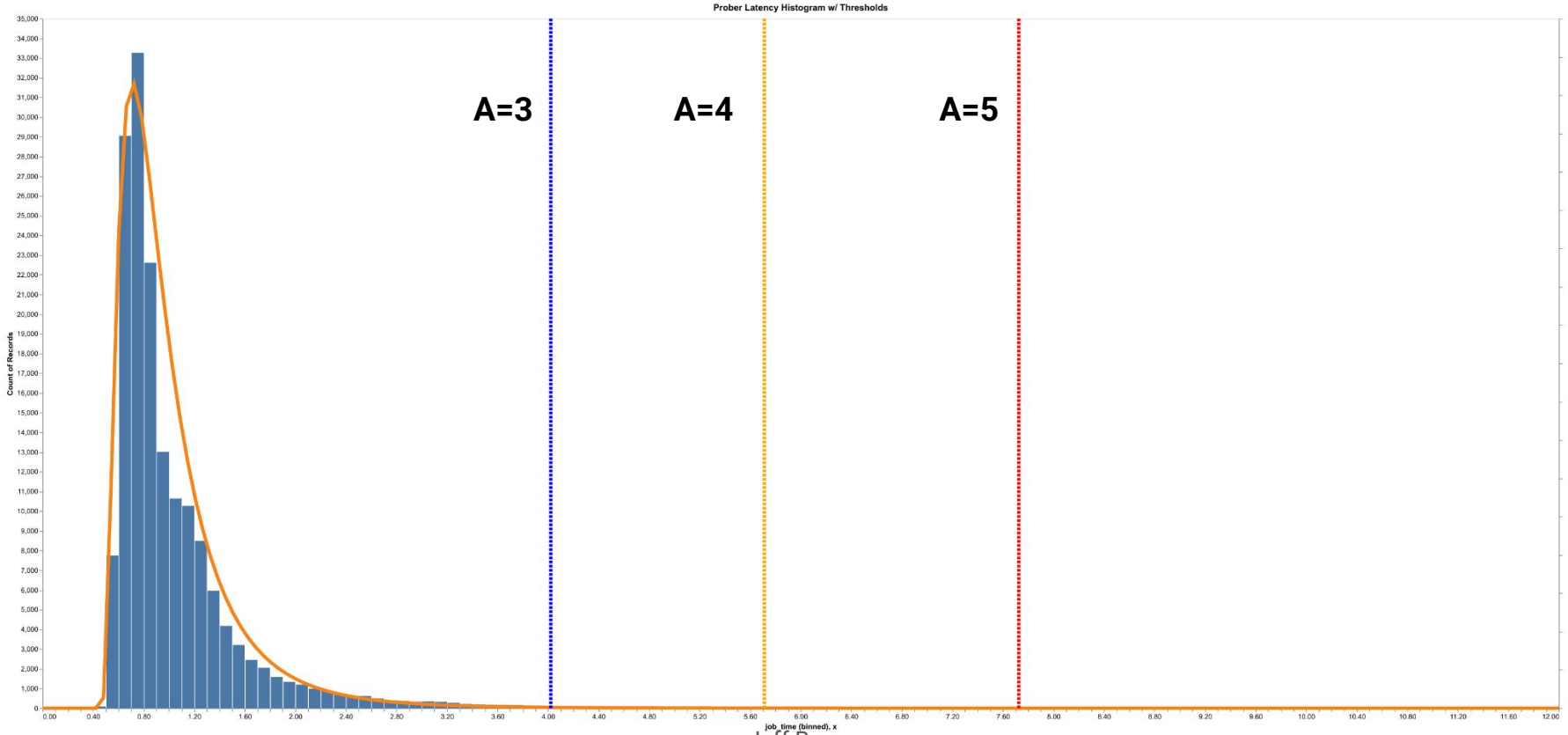
An Alert?


Probe Time (seconds)
100
95
90
85
80
75
70
65
60
55
50
45
40
35
30
25
20
15
10
5
0
Time

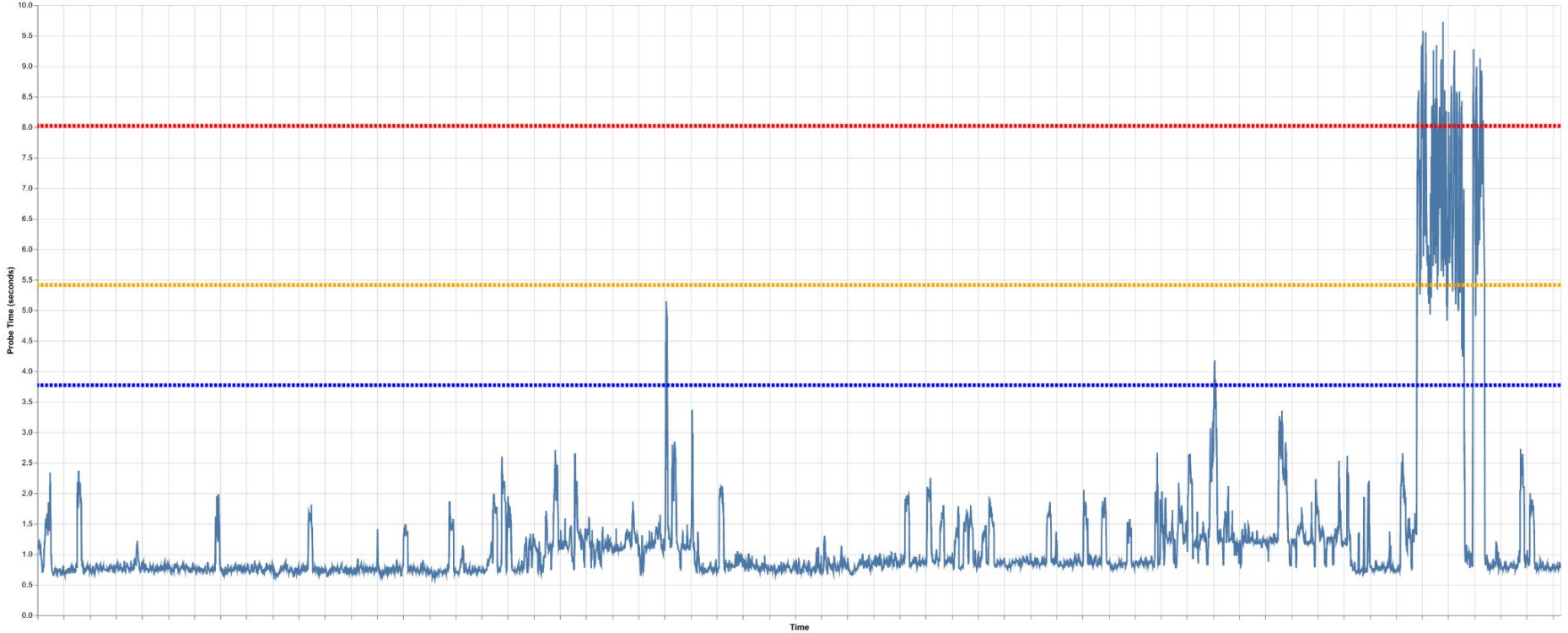
Another Example



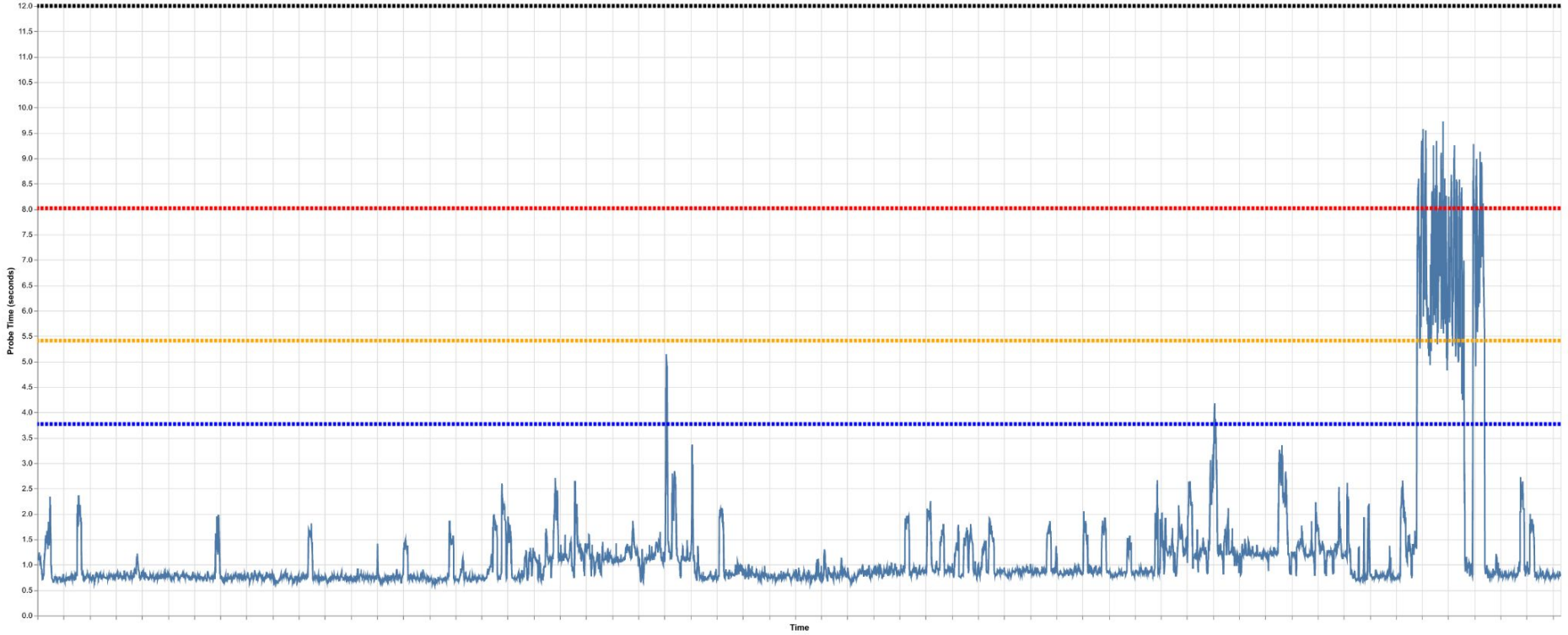
Another Example



An Alert!


Probe Time (seconds)
10.0
9.5
9.0
8.5
8.0
7.5
7.0
6.5
6.0
5.5
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Time

An Alert!


Probe Time (seconds)
12.0
11.5
11.0
10.5
10.0
9.5
9.0
8.5
8.0
7.5
7.0
6.5
6.0
5.5
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Time

Conclusions

Conclusions

- Not all environments are the same!
- Probers shouldn’t be pets
- Using simple statistical techniques lets us:
  - Get more out of our existing probers
  - Eliminate toil
  - Still provide knobs for tuning