Do You Really Know Your Response Times?

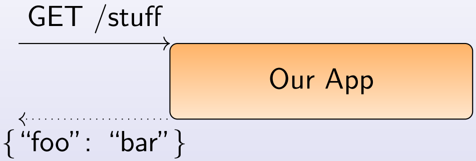
Daniel Rolls

March 2017

Sky Over The Top Delivery

- Web Services
- Over The Top Asset Delivery
- NowTV/Sky Go
- Always up
- High traffic
- Highly concurrent

OTT Endpoints

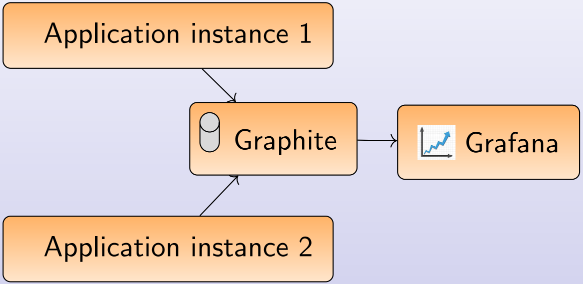


- How much traffic is hitting that endpoint?
- How quickly are we responding to a typical customer?
- One customer complained we respond slowly. How slow do we get?
- What's the difference between the fastest and slowest responses?
- I don't care about anomalies but how slow are the slowest 1%?

Collecting Response Times

- Large volumes of network traffic
- Risk of losing data (network may fail)
- Affects application performance
- Needs measuring itself!

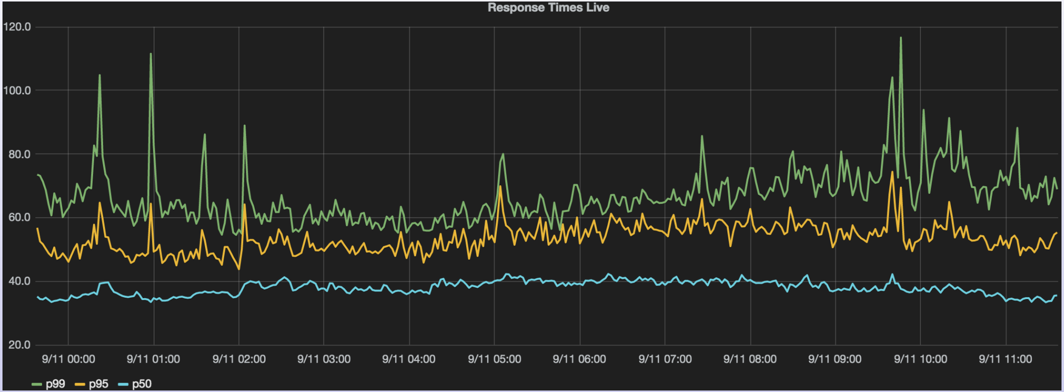
Our Setup


Application instance 1
Graphite
Grafana
Application instance 2

Dropwizard Metrics Library: Types of Metric

- Counter
- Gauge — 'instantaneous measurement of a value'
- Meter (counts, rates)
- Histogram — min, max, mean, stddev, percentiles
- Timer — Meter + Histogram

Example Dashboard



Dropwizard Metrics

- Use Dropwizard and you get
  - Metrics infrastructure for free
  - Metrics from Cassandra and Dropwizard bundles for free
  - You can easily add timers to metrics just by adding annotations
- Ports exist for other languages
- Developers, architects, managers everybody loves graphs
- We trust and depend on them
- We rarely understand them
- We lie to ourselves and to our managers with them

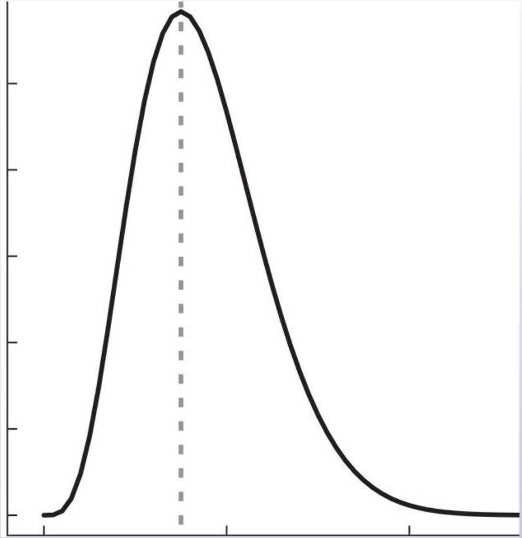
Goals of this talk

- Understand how we can measure service time latencies
- Ensure meaningful statistics are given back to managers
- Learn how to use appropriate dashboards for monitoring and alerting

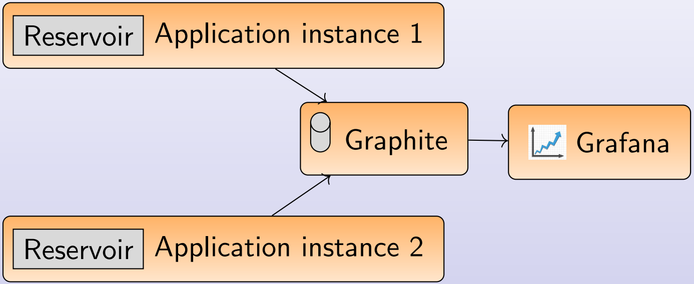
What is the 99th Percentile Response Time?

?

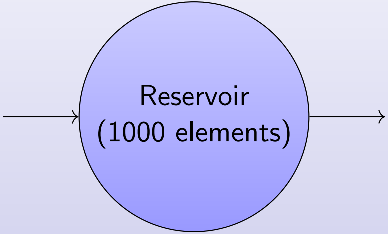
What is the 99^{th} Percentile?



Our Setup


Reservoir
Application instance 1
Graphite
Grafana
Reservoir
Application instance 2

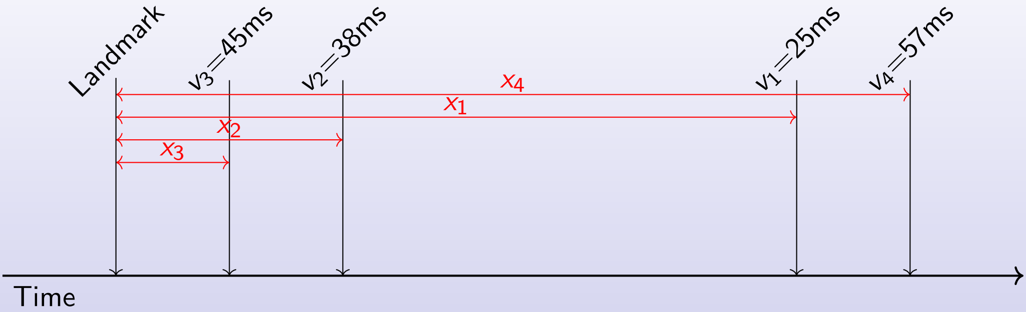
Reservoirs



Types of Reservoir

- Sliding window
- Time-base sliding window
- Exponentially decaying

Forward Decay



$$w_i = e^{\alpha x_i}$$

w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8
v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8

$\longrightarrow$

Sorted by value

Getting at the percentiles

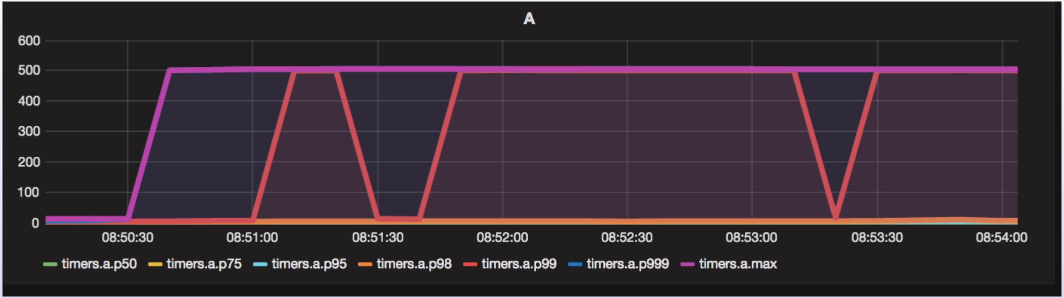
- Normalise weights: $\sum_i w_i = 1$
- Lookup by normalised weight

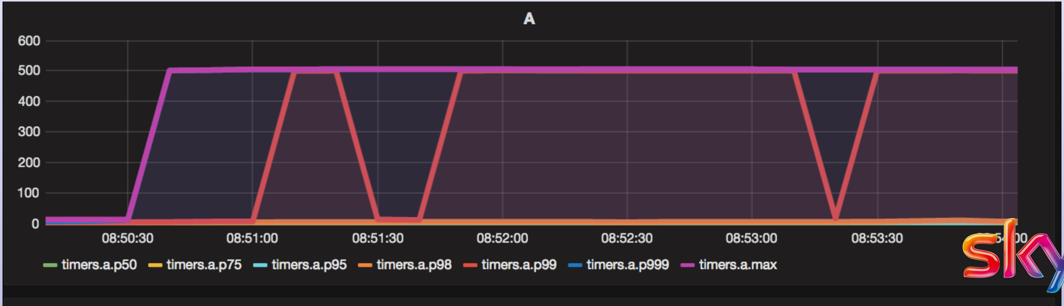
Data retention

- Sorted Map indexed by w.random_number
- Smaller indices removed first

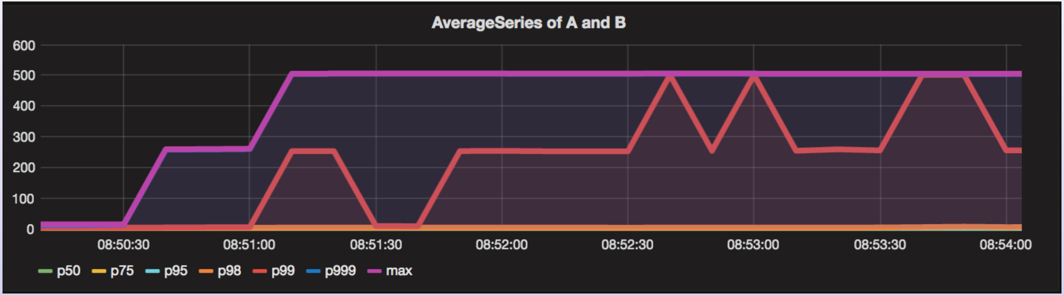
Response Time Jumps for 4 Minutes

One Percent Rise from 20ms to 500ms





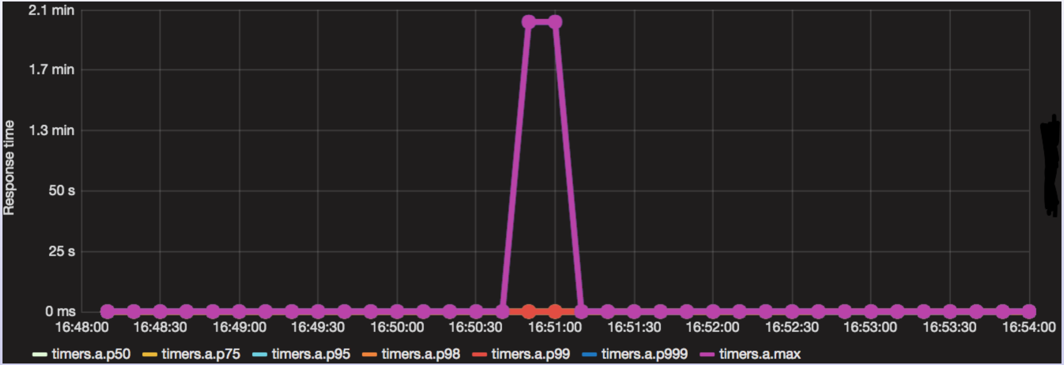
One Percent Rise from 20ms to 500ms



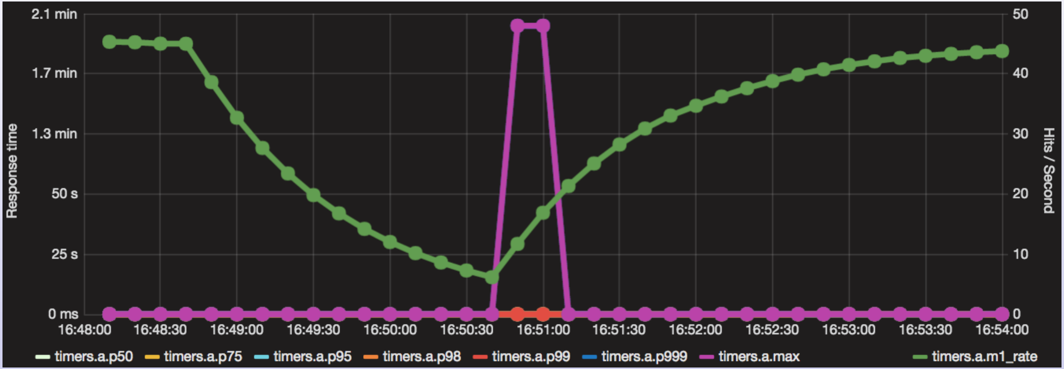
Trade-off

- Autonomous teams
  - Know one app well
  - Feel responsible for app performance
- But. . .
  - Can't know everything
  - Will make mistakes with numbers
  - We might even ignore mistakes

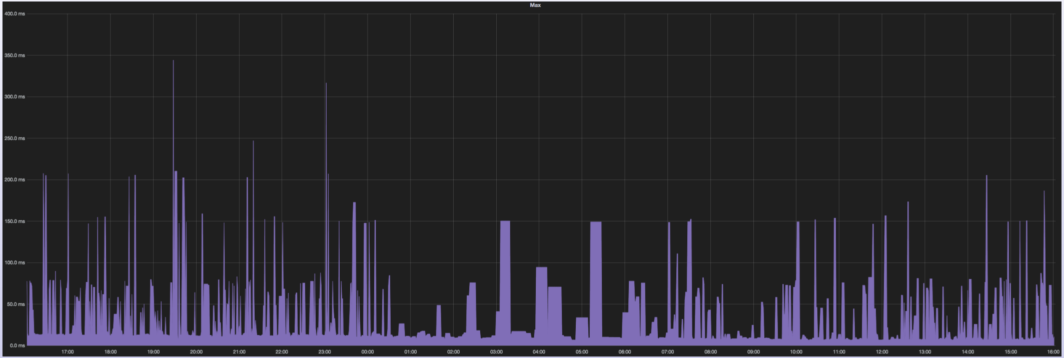
One Long Request Blocks New Requests



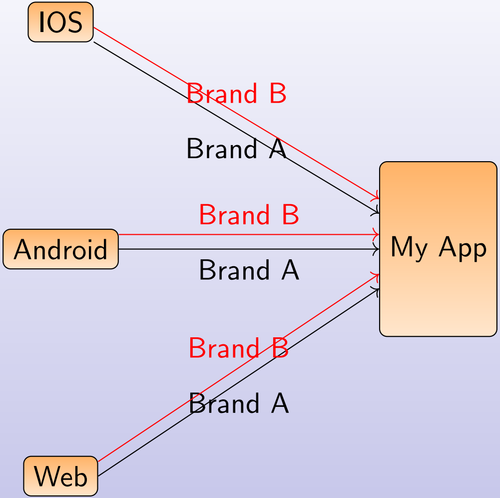
One Long Request Blocks New Requests



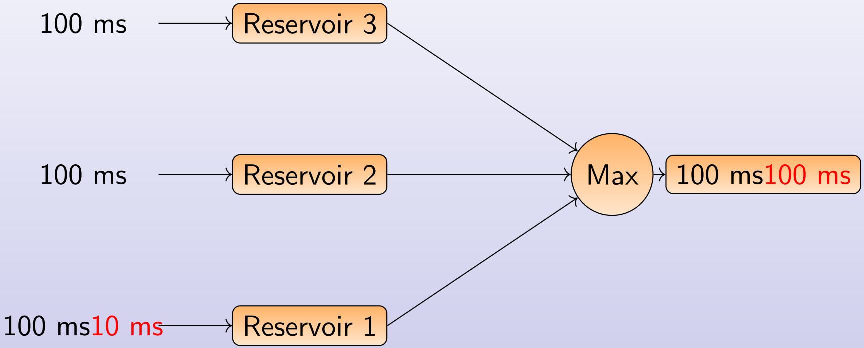
Spikes and Tower Blocks



Splitting Things Up



Metric Imbalance Visualised



Metric Imbalance

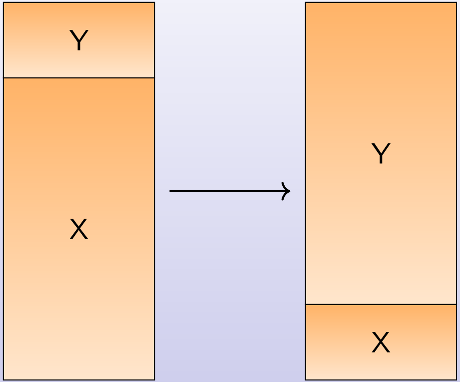
- One pool gives more accurate results
- Multiple pools allow drilling down, but. . .
  - Some pools may have inaccurate performance measurements
  - Only those with sufficient rates should be analysed
  - How can we narrow down on just those?
- Simpson's Paradox

Simpson's Paradox

Explanation

- Two variables have a positive correlation
- Grouped data shows a negative correlation
- There's a lurking third variable

Simpson's Paradox



- Increasing traffic $\implies$ X gets slower
- Increasing traffic $\implies$ Y gets faster
- We move % traffic to System Y
- We wait for prime time peak
- System gets slower???
- 100% of brand B traffic still goes to X
- Results are pooled by client and brand
- Classic example: UC Berkeley gender bias

Lessons Learnt

- Want fast alerting?
  - Use max
    - If you don't graph the max you are hiding the bad
- Don't just look at fixed percentiles.
  - Understand the distribution of the data (HdrHistogram)
  - A few fixed percentiles tells you very little as a test
- Monitor one metric per endpoint
- When aggregating response times
  - Use maxSeries

So We're Living a Lie, Does it Matter?

Conclusions and Thoughts

- Don't immediately assume numbers on dashboards are meaningful
- Understand what you are graphing
- Test assumptions
- Provide these tools and developers will confidently use them
  - Although maybe not correctly!
  - Most developers are not mathematicians
- Keep it simple!
- Know which numbers are real and which are lies!

Thank you