# Three Pillars with Zero Answers

A New Observability Scorecard

December 11, 2018

LIGHTSTEP

## Part I

## A Critique

## The Conventional Wisdom

Observing microservices is hard

Google and Facebook solved this (right???)

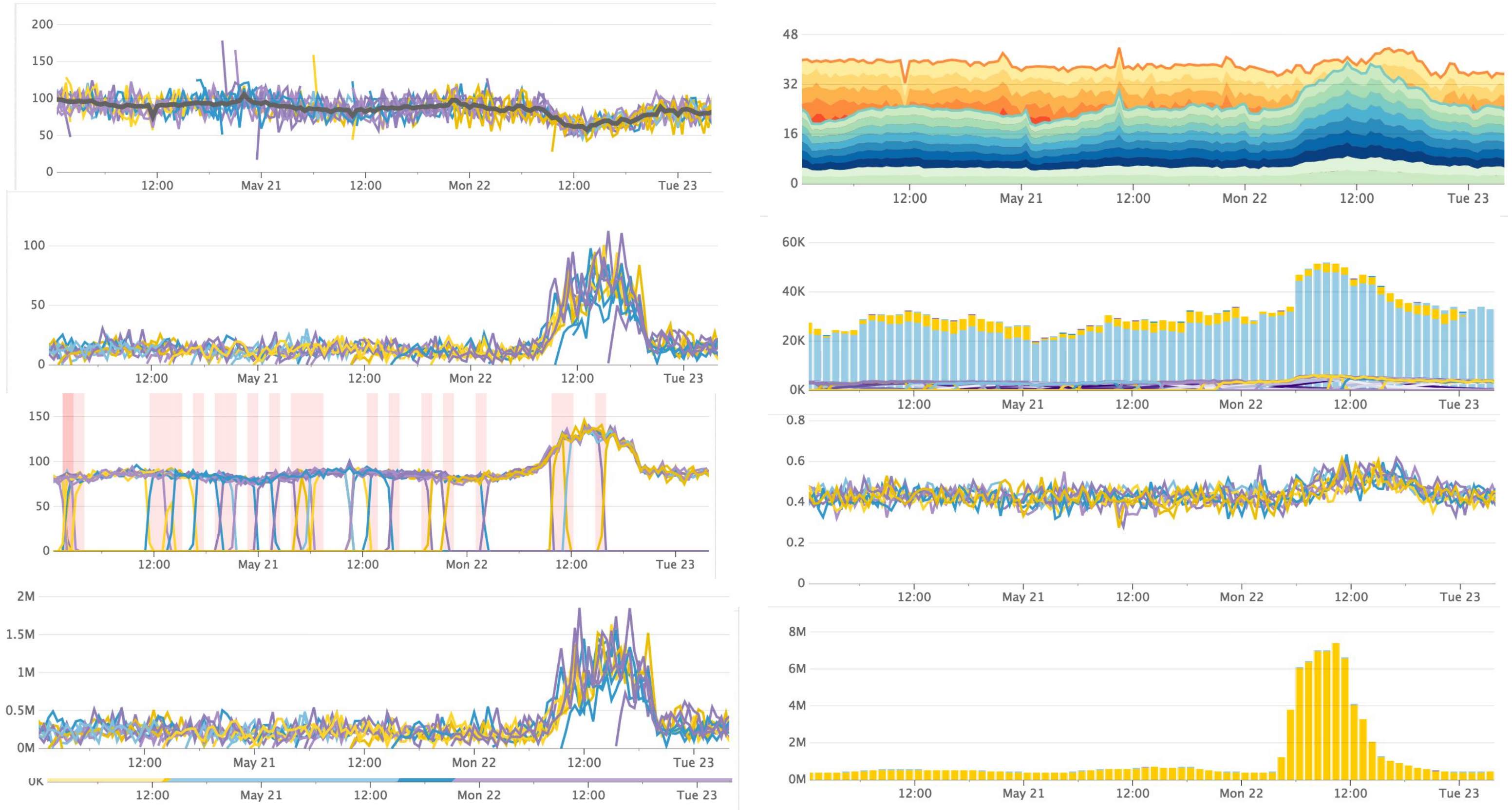
They used **Metrics, Logging, and Distributed Tracing**…

So we should, too.

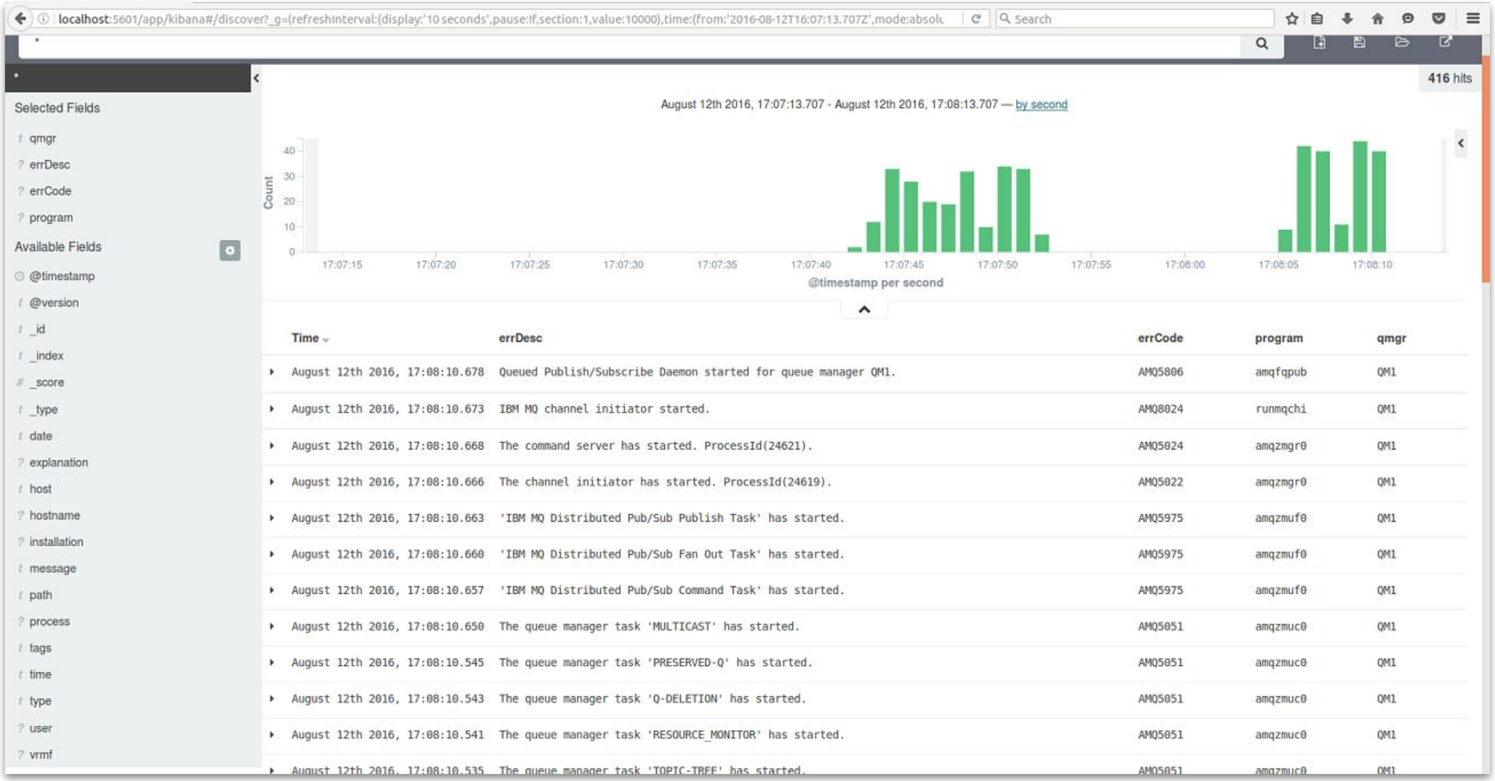
## The Three Pillars of Observability

- Metrics
- Logging
- Distributed Tracing

# Metrics!

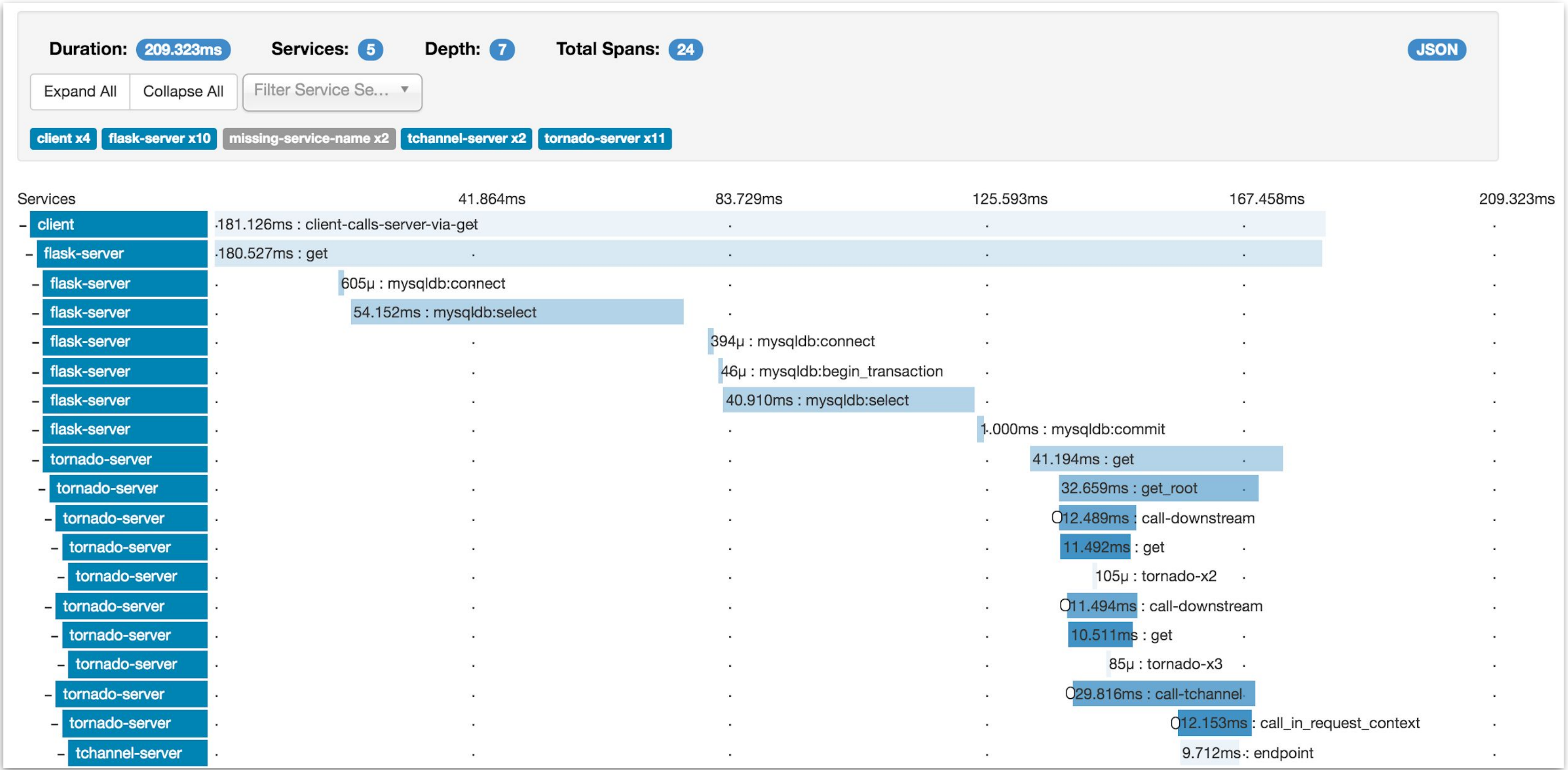

200
150
100
50
0
12:00
May 21
Mon 22
Tue 23
48
32
16
60K
40K
20K
0K
0.8
0.6
0.4
0.2
2M
1.5M
1M
0.5M
0M
8M
6M
4M
2M

# Logging!



| Time | errDesc | errCode | program | qmgr |
|---|---|---|---|---|
| August 12th 2016, 17:08:10.678 | Queued Publish/Subscribe Daemon started for queue manager QM1. | AMQ5806 | amqfqpub | QM1 |
| August 12th 2016, 17:08:10.673 | IBM MQ channel initiator started. | AMQ8024 | runmqchi | QM1 |
| August 12th 2016, 17:08:10.668 | The command server has started. ProcessId(24621). | AMQ5024 | amqzmgr0 | QM1 |
| August 12th 2016, 17:08:10.666 | The channel initiator has started. ProcessId(24619). | AMQ5022 | amqzmgr0 | QM1 |
| August 12th 2016, 17:08:10.663 | 'IBM MQ Distributed Pub/Sub Publish Task' has started. | AMQ5975 | amqzmuf0 | QM1 |
| August 12th 2016, 17:08:10.660 | 'IBM MQ Distributed Pub/Sub Fan Out Task' has started. | AMQ5975 | amqzmuf0 | QM1 |
| August 12th 2016, 17:08:10.657 | 'IBM MQ Distributed Pub/Sub Command Task' has started. | AMQ5975 | amqzmuf0 | QM1 |
| August 12th 2016, 17:08:10.650 | The queue manager task 'MULTICAST' has started. | AMQ5051 | amqzmuc0 | QM1 |
| August 12th 2016, 17:08:10.545 | The queue manager task 'PRESERVED-Q' has started. | AMQ5051 | amqzmuc0 | QM1 |
| August 12th 2016, 17:08:10.543 | The queue manager task 'Q-DELETION' has started. | AMQ5051 | amqzmuc0 | QM1 |
| August 12th 2016, 17:08:10.541 | The queue manager task 'RESOURCE_MONITOR' has started. | AMQ5051 | amqzmuc0 | QM1 |
| August 12th 2016, 17:08:10.535 | The queue manager task 'TOPIC-TREE' has started. | AMQ5051 | amqzmuc0 | QM1 |

# Tracing!



Duration: 209.323ms Services: 5 Depth: 7 Total Spans: 24 JSON

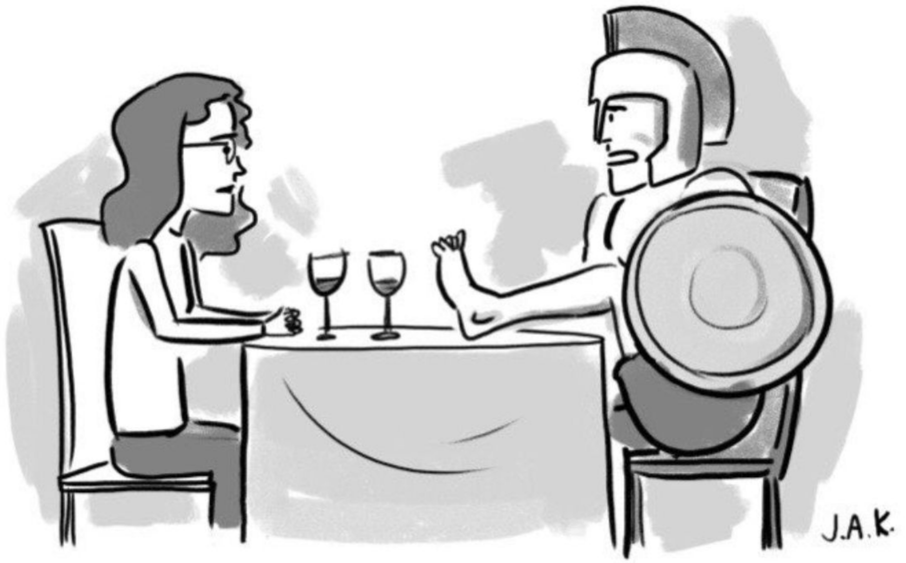
Expand All | Collapse All | Filter Service Se...

client x4 | flask-server x10 | missing-service-name x2 | tchannel-server x2 | tornado-server x11

```
Services          41.864ms     83.729ms     125.593ms     167.458ms     209.323ms
- client          181.126ms : client-calls-server-via-get
 - flask-server   180.527ms : get
  - flask-server  605μ : mysqldb:connect
  - flask-server  54.152ms : mysqldb:select
  - flask-server  394μ : mysqldb:connect
  - flask-server  46μ : mysqldb:begin_transaction
  - flask-server  40.910ms : mysqldb:select
  - flask-server  1.000ms : mysqldb:commit
  - tornado-server  41.194ms : get
   - tornado-server  32.659ms : get_root
    - tornado-server  12.489ms : call-downstream
     - tornado-server  11.492ms : get
      - tornado-server  105μ : tornado-x2
    - tornado-server  11.494ms : call-downstream
     - tornado-server  10.511ms : get
      - tornado-server  85μ : tornado-x3
    - tornado-server  29.816ms : call-tchannel
     - tornado-server  12.153ms : call_in_request_context
      - tchannel-server  9.712ms : endpoint
```

POP
SNAP
CRACKLE
CRACKLE!
POP!

## Fatal Flaws

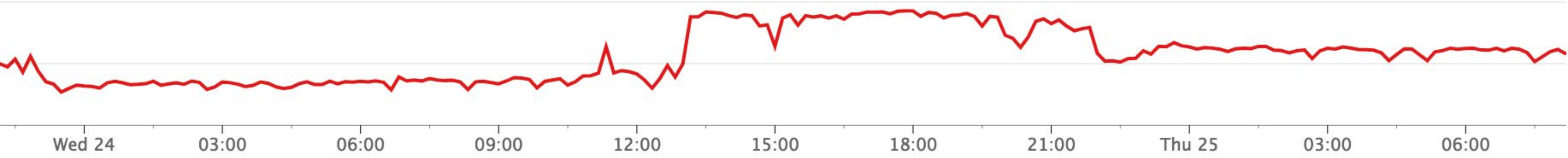


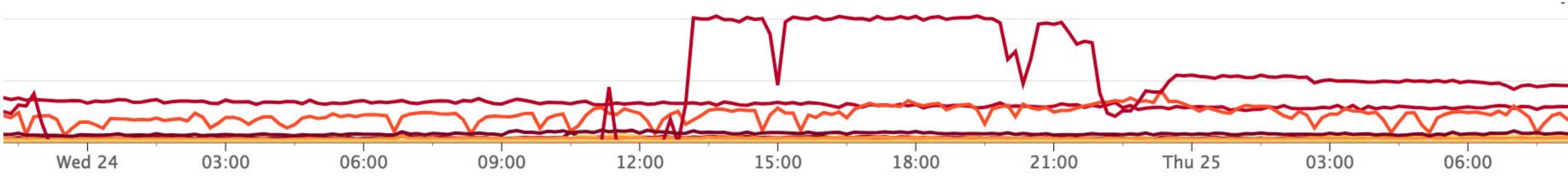
"I'm ready to be vulnerable."

# A word nobody knew in 2015…

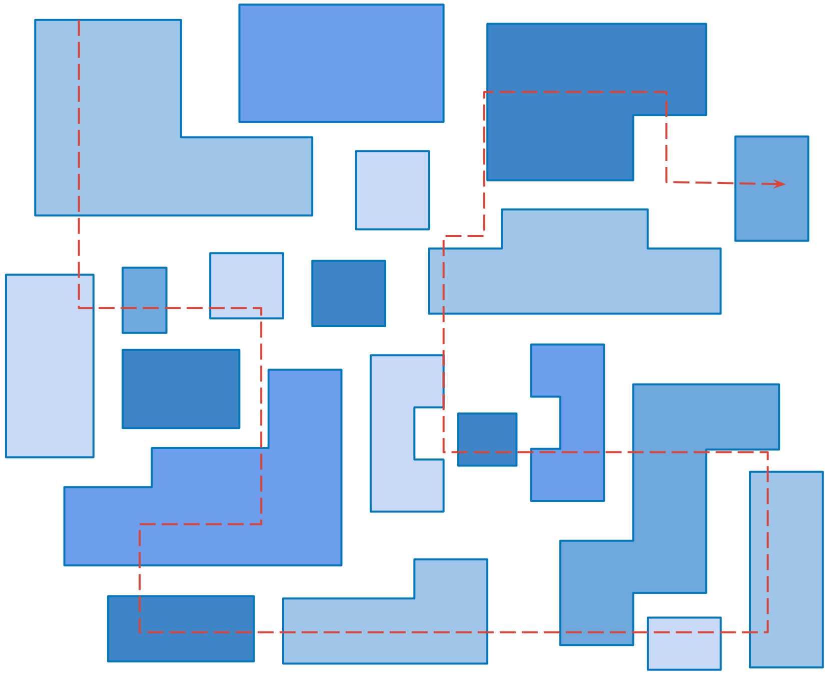
Dimensions (aka “tags”) can explain variance in timeseries data (aka “metrics”) …

… but **cardinality**





## Logging Data Volume: a reality check



```
        transaction rate
x     all microservices
x   cost of net+storage
x    weeks of retention
-----------------------
```

***way*** `too much $$$$`

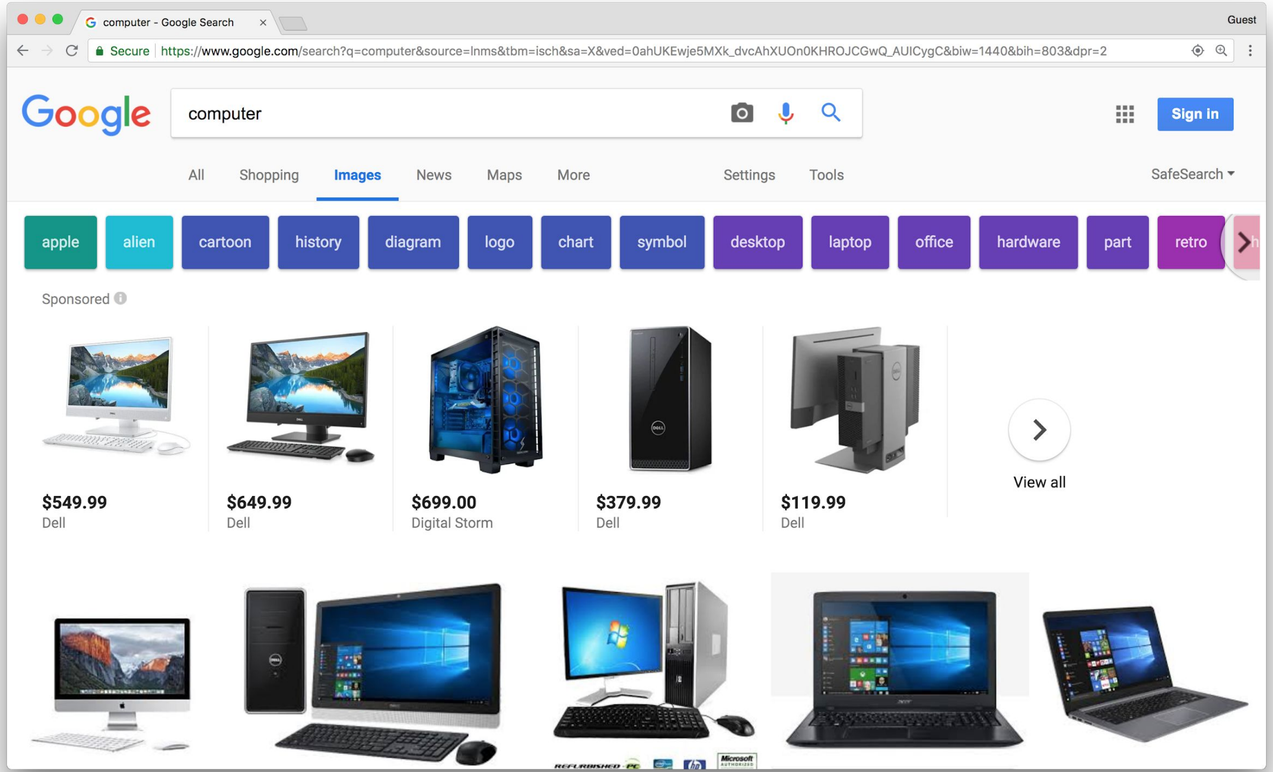
## The Life of Transaction Data: Dapper

| **Stage** | **Overhead affects…** | **Retained** |
|---|---|---|
| Instrumentation Executed | App | 100.00% |
| Buffered within app process | App | 000.10% |
| Flushed out of process | App | 000.10% |
| Centralized regionally | Regional network + storage | 000.10% |
| Centralized globally | WAN + storage | 000.01% |

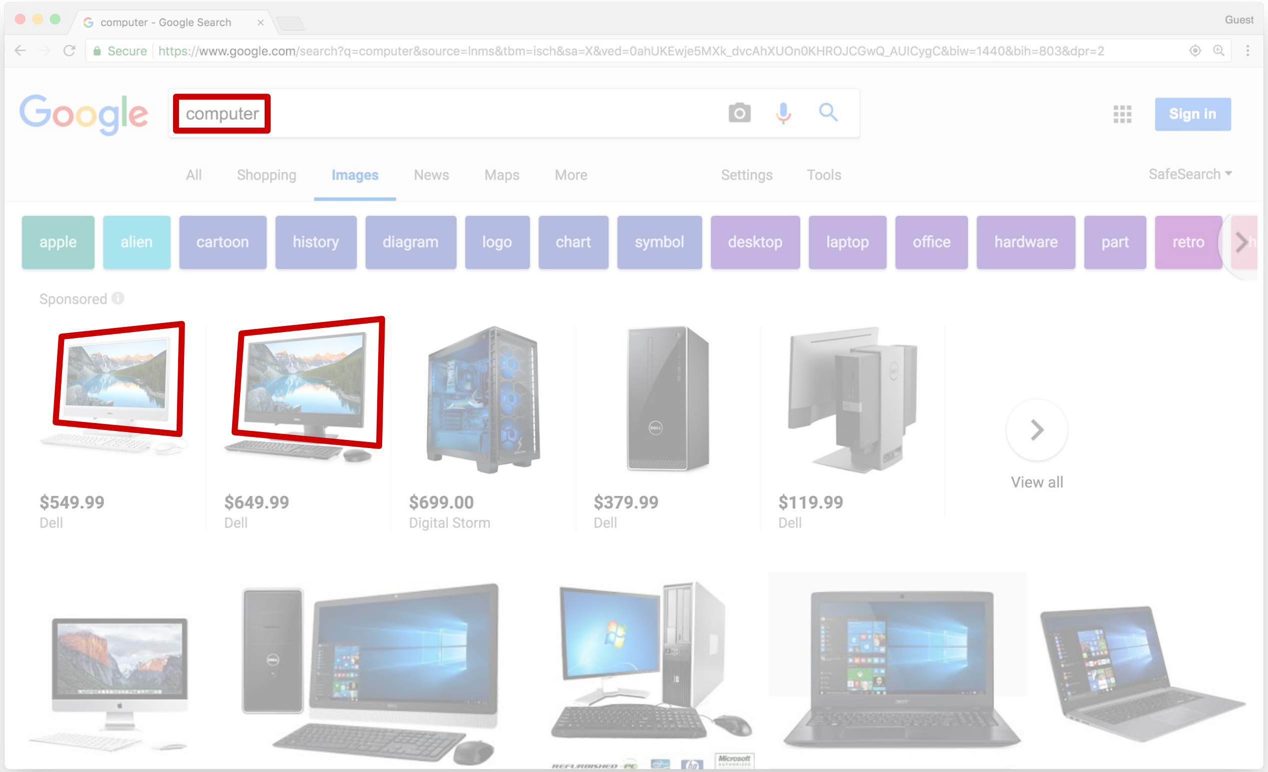
# Fatal Flaws: A Review

| | Logs | Metrics | Dist. Traces |
|---|---|---|---|
| TCO scales gracefully | — | ✓ | ✓ |
| Accounts for all data (i.e., unsampled) | ✓ | ✓ | — |
| Immune to cardinality | ✓ | — | ✓ |

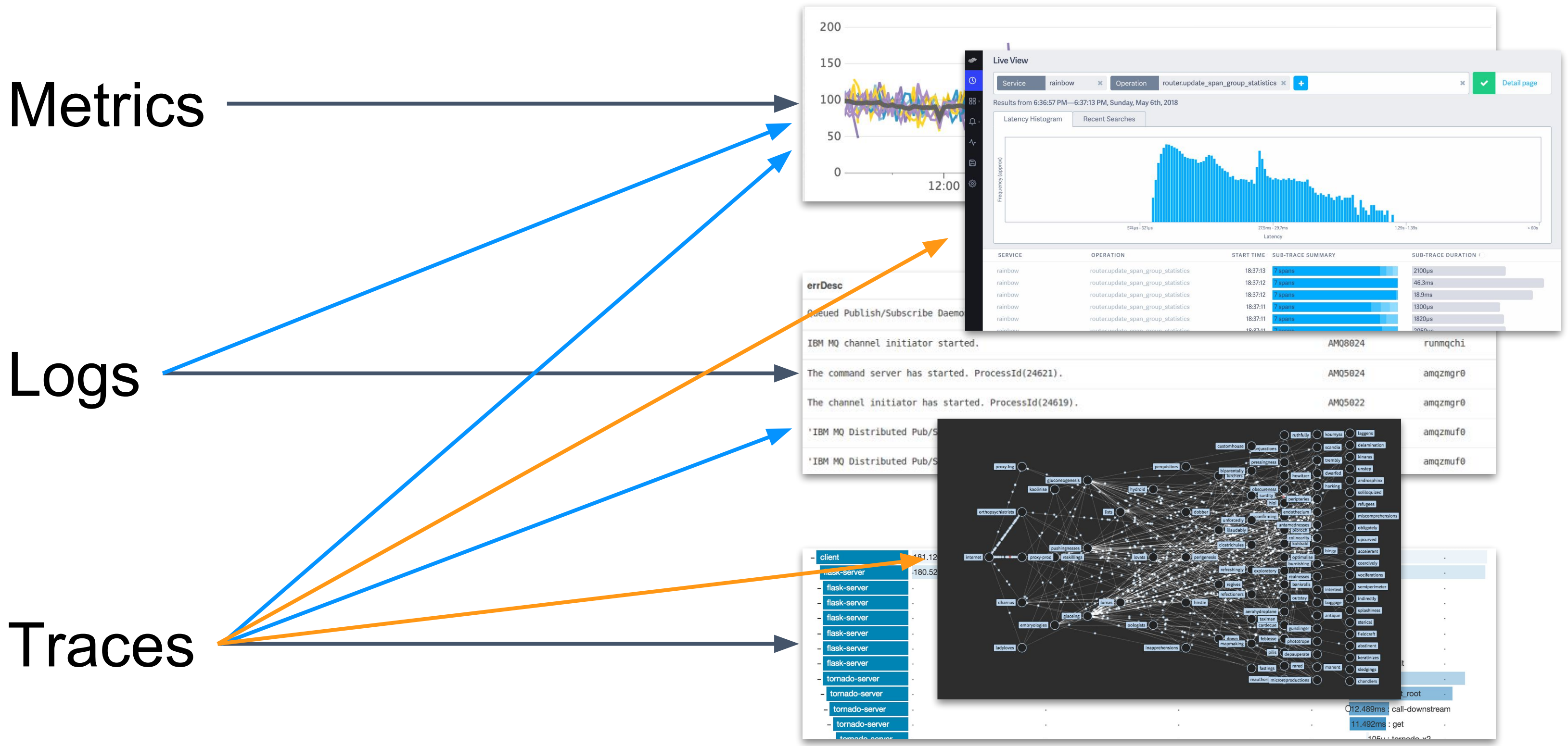
# Data vs UI


computer - Google Search
Guest
Secure
https://www.google.com/search?q=computer&source=lnms&tbm=isch&sa=X&ved=0ahUKEwje5MXk_dvcAhXUOn0KHROJCGwQ_AUICygC&biw=1440&bih=803&dpr=2
Google
computer
Sign in
All
Shopping
Images
News
Maps
More
Settings
Tools
SafeSearch
apple
alien
cartoon
history
diagram
logo
chart
symbol
desktop
laptop
office
hardware
part
retro
Sponsored
$549.99
Dell
$649.99
Dell
$699.00
Digital Storm
$379.99
Dell
$119.99
Dell
View all

# Data vs UI


computer - Google Search
Guest
Secure https://www.google.com/search?q=computer&source=lnms&tbm=isch&sa=X&ved=0ahUKEwje5MXk_dvcAhXUOn0KHROJCGwQ_AUICygC&biw=1440&bih=803&dpr=2
Google
computer
Sign in
All
Shopping
Images
News
Maps
More
Settings
Tools
SafeSearch
apple
alien
cartoon
history
diagram
logo
chart
symbol
desktop
laptop
office
hardware
part
retro
Sponsored
$549.99
Dell
$649.99
Dell
$699.00
Digital Storm
$379.99
Dell
$119.99
Dell
View all

# Data vs UI



Service rainbow · Operation router.update_span_group_statistics

Results from 6:36:57 PM—6:37:13 PM, Sunday, May 6th, 2018

| SERVICE | OPERATION | START TIME | SUB-TRACE SUMMARY | SUB-TRACE DURATION |
|---|---|---|---|---|
| rainbow | router.update_span_group_statistics | 18:37:13 | 7 spans | 2100µs |
| rainbow | router.update_span_group_statistics | 18:37:12 | 7 spans | 46.3ms |
| rainbow | router.update_span_group_statistics | 18:37:12 | 7 spans | 18.9ms |
| rainbow | router.update_span_group_statistics | 18:37:11 | 7 spans | 1300µs |
| rainbow | router.update_span_group_statistics | 18:37:11 | 7 spans | 1820µs |

```
Queued Publish/Subscribe Daemo
IBM MQ channel initiator started.                              AMQ8024     runmqchi
The command server has started. ProcessId(24621).              AMQ5024     amqzmgr0
The channel initiator has started. ProcessId(24619).           AMQ5022     amqzmgr0
'IBM MQ Distributed Pub/S                                                  amqzmuf0
'IBM MQ Distributed Pub/S                                                  amqzmuf0
```

```
client          181.12
flask-server    180.52
12.489ms : call-downstream
11.492ms : get
```

Metrics, Logs, and Traces are **Just Data,**

… not a feature or use case.

LIGHTSTEP

## Part II

## A New Scorecard for Observability

## Mental Model: Goals and Activities

**Goals:** how our services perform in the eyes of their consumers

**Activities:** what we (as operators) actually *do* to further our *goals*

## Quick Vocab Refresher: **SLIs**

“SLI” = “Service Level Indicator”

TL;DR: An SLI is **an indicator of health** that a service’s **consumers** would care about.

… *not* an indicator of its inner workings

LIGHTSTEP

## Observability: Two Fundamental **Goals**

- Gradually **improving** an SLI
- Rapidly **restoring** an SLI

days, weeks, months…

**NOW!!!!**

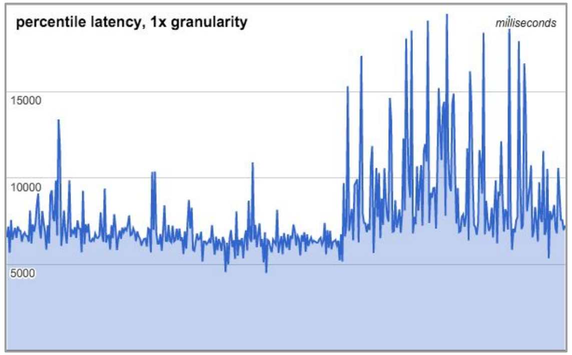
Reminder: “SLI” = “Service Level Indicator”

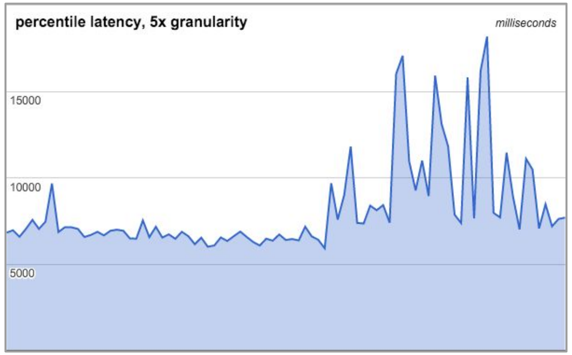
## Observability: Two Fundamental **Activities**

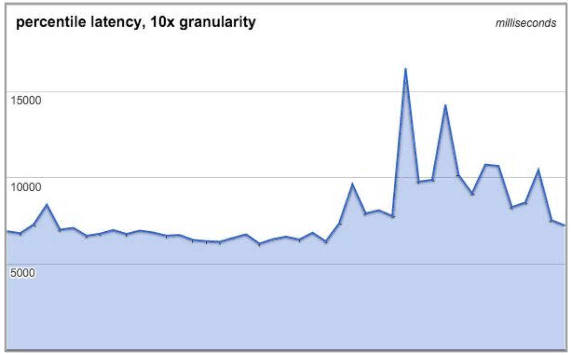
1. **Detection:** measuring SLIs precisely
2. **Refinement:** reducing the search space for plausible explanations

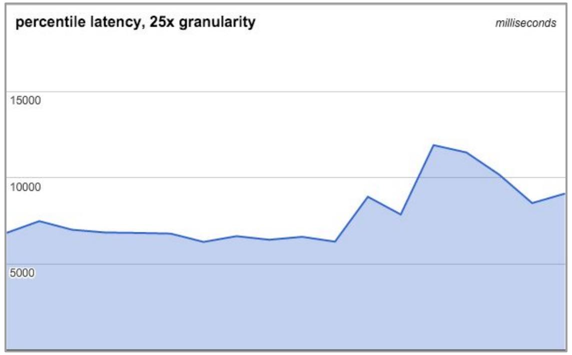
LIGHTSTEP

# An interlude about stats frequency









## Scorecard: **Detection**

Specificity:

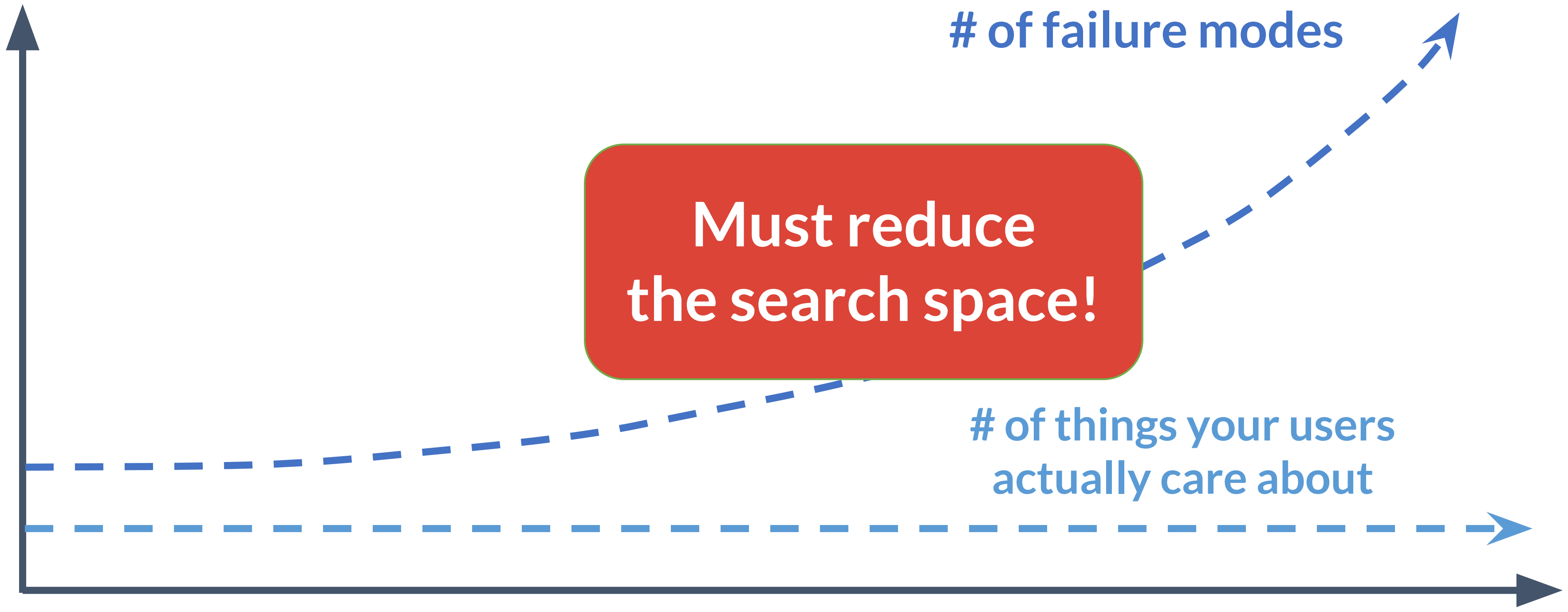
- Cost of cardinality (*$ per tag value*)
- Stack support (*mobile/web platforms, managed services, "black-box OSS infra" like Kafka/Cassandra*)

Fidelity:

- Correct stats!!! (*global p95, p99*)
- High stats frequency (*stats sampling frequency, in seconds*)

Freshness: (*lag from real-time, in seconds*)

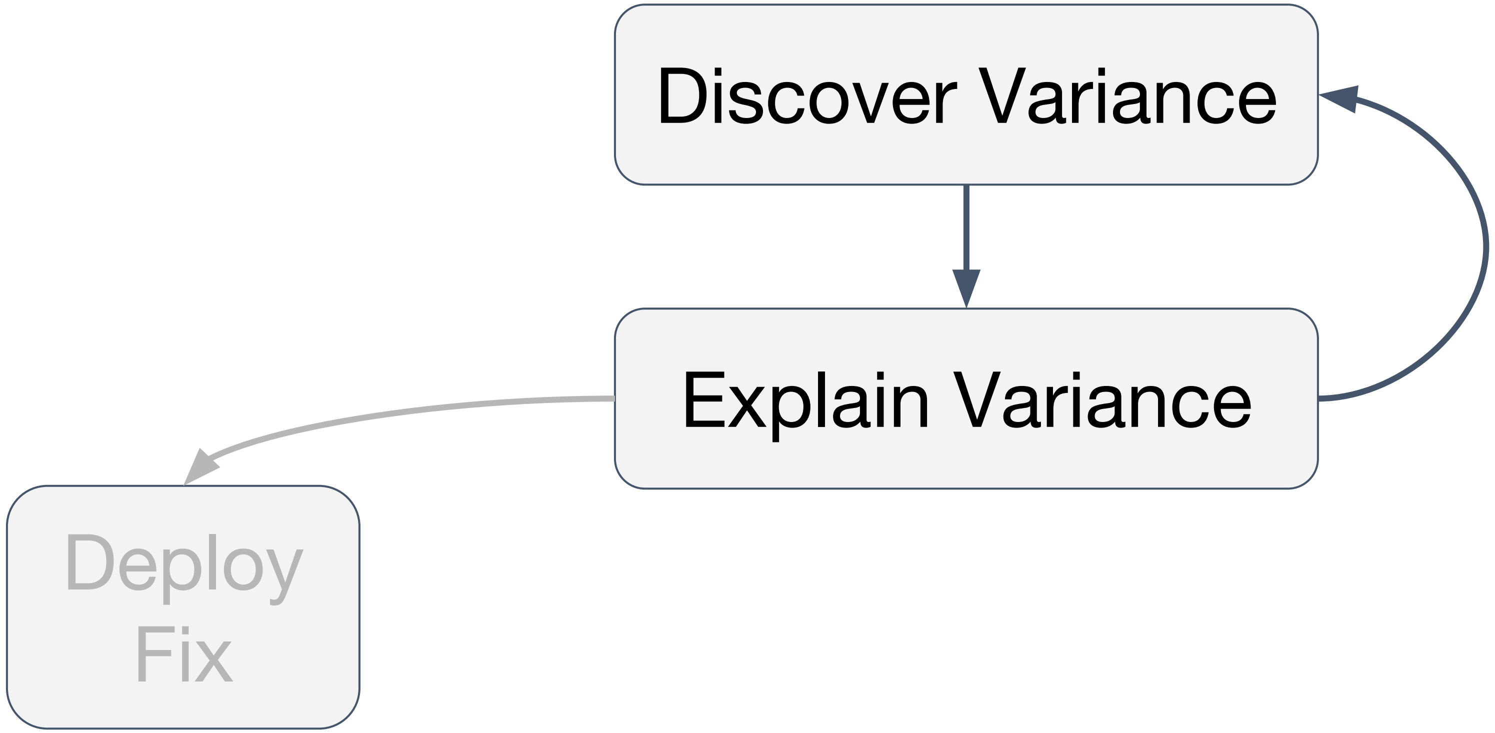
## Why “Refinement”?



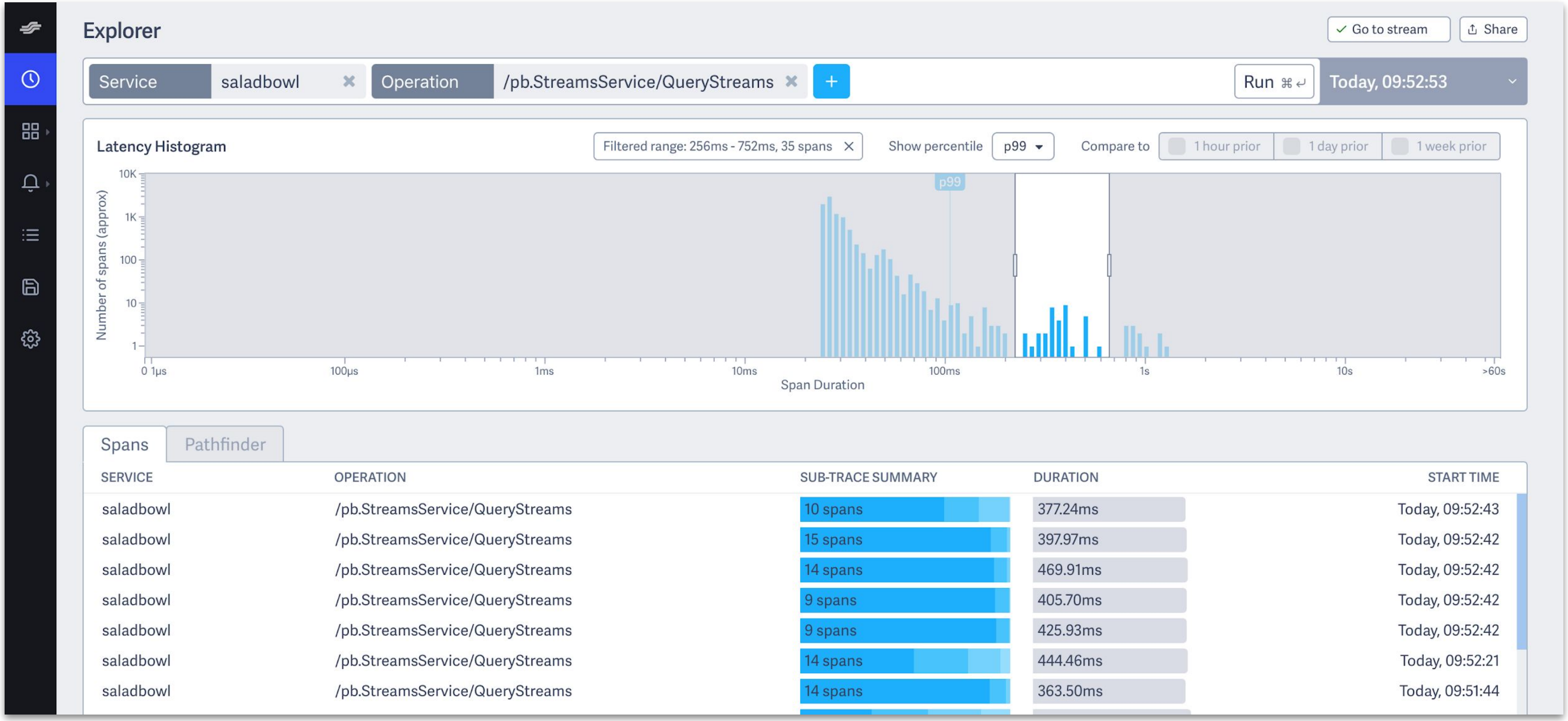
Must reduce the search space!

# of microservices

## The Refinement Process



# Histograms vs “p99”



| SERVICE | OPERATION | SUB-TRACE SUMMARY | DURATION | START TIME |
|---|---|---|---|---|
| saladbowl | /pb.StreamsService/QueryStreams | 10 spans | 377.24ms | Today, 09:52:43 |
| saladbowl | /pb.StreamsService/QueryStreams | 15 spans | 397.97ms | Today, 09:52:42 |
| saladbowl | /pb.StreamsService/QueryStreams | 14 spans | 469.91ms | Today, 09:52:42 |
| saladbowl | /pb.StreamsService/QueryStreams | 9 spans | 405.70ms | Today, 09:52:42 |
| saladbowl | /pb.StreamsService/QueryStreams | 9 spans | 425.93ms | Today, 09:52:42 |
| saladbowl | /pb.StreamsService/QueryStreams | 14 spans | 444.46ms | Today, 09:52:21 |
| saladbowl | /pb.StreamsService/QueryStreams | 14 spans | 363.50ms | Today, 09:51:44 |

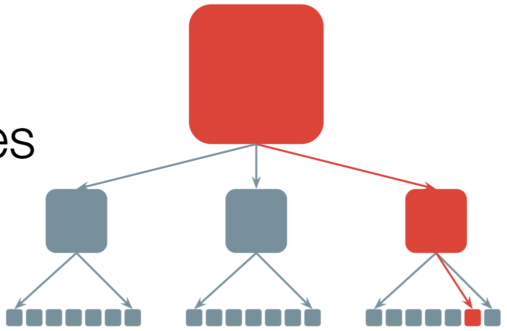
# Scorecard: **Refinement**

Identifying Variance:

- Cardinality (*$ per tag value*)
- Robust stats (*histograms* (see prev slide))
- Retention horizons for plausible queries (*time duration*)

Explaining variance:

- Correct stats!!! (*global p95, p99*)
- “Suppress the messengers” of microservice failures



## Wrapping up…

(first, a hint at my perspective)

(Review)

# The Life of Trace Data: Dapper

| **Stage** | **Overhead affects…** | **Retained** |
|---|---|---|
| Instrumentation Executed | App | 100.00% |
| Buffered within app process | App | 000.10% |
| Flushed out of process | App | 000.10% |
| Centralized regionally | Regional network + storage | 000.10% |
| Centralized globally | WAN + storage | 000.01% |

LIGHTSTEP

# The Life of Trace Data: ~~Dapper~~ Other Approaches

| **Stage** | **Overhead affects…** | **Retained** |
|---|---|---|
| Instrumentation Executed | App | 100.00% |
| Buffered within app process | App | 100.00% |
| Flushed out of process | App | 100.00% |
| Centralized regionally | Regional network + storage | 100.00% |
| Centralized globally | WAN + storage | on-demand |

LIGHTSTEP

## An Observability Scorecard

**Detection**

- Specificity: cardinality cost, stack coverage
- Fidelity: correct stats, high stats frequency
- Freshness: ≤ 5 seconds

**Refinement**

- Identifying variance: cardinality cost, correct stats, hi-fi histograms, retention horizons
- “Suppress the messengers”

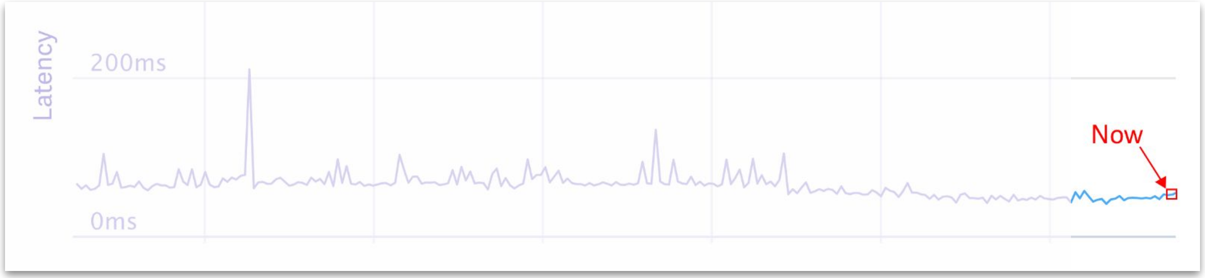
# Thank you!

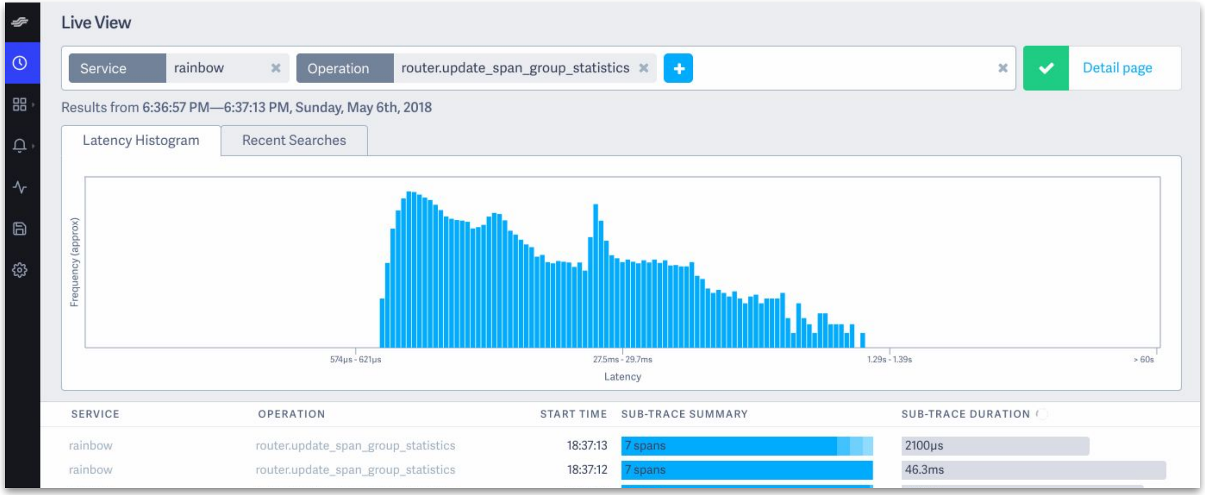
Ben Sigelman, Co-founder and CEO
twitter: @el_bhs
email: bhs@lightstep.com

LIGHTSTEP

## Extra slides

# Ideal Measurement: Robust





Live View

Service rainbow | Operation router.update_span_group_statistics | Detail page

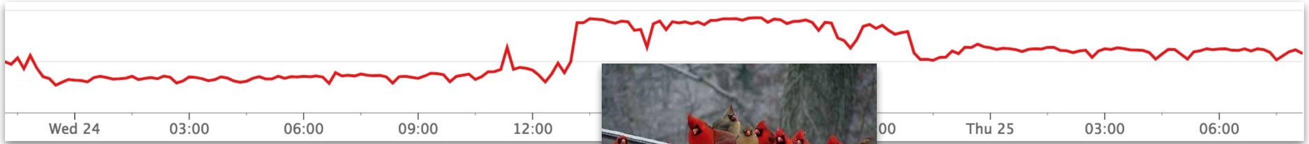
Results from 6:36:57 PM—6:37:13 PM, Sunday, May 6th, 2018

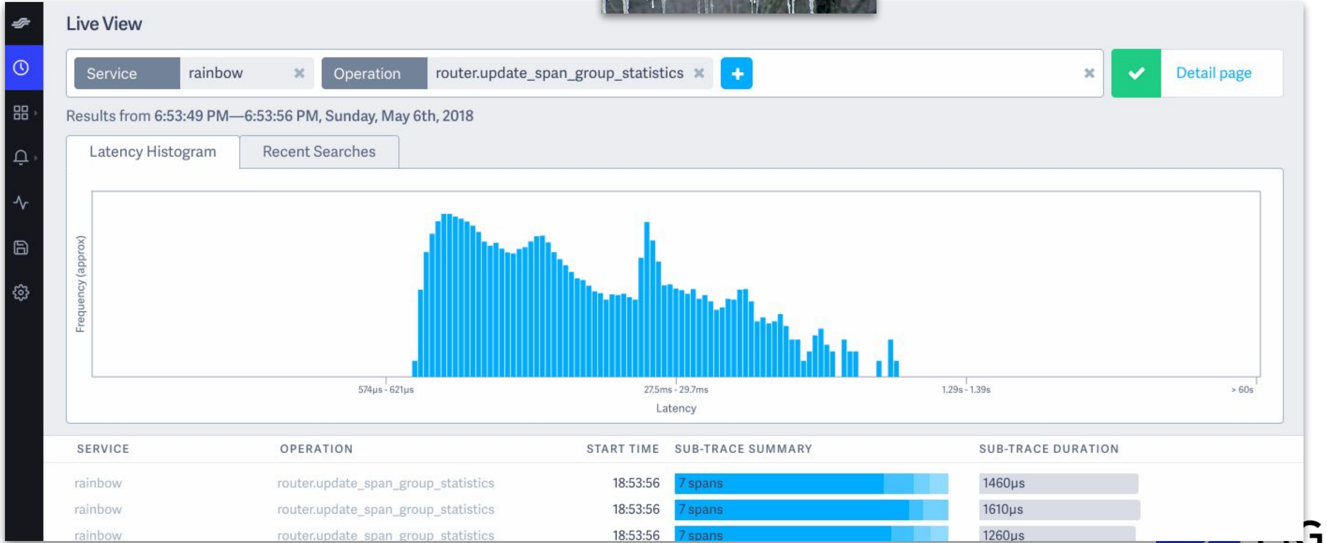
Latency Histogram | Recent Searches

| SERVICE | OPERATION | START TIME | SUB-TRACE SUMMARY | SUB-TRACE DURATION |
|---|---|---|---|---|
| rainbow | router.update_span_group_statistics | 18:37:13 | 7 spans | 2100µs |
| rainbow | router.update_span_group_statistics | 18:37:12 | 7 spans | 46.3ms |

LIGHTSTEP

# Ideal Measurement: High-Dimensional





Results from 6:53:49 PM—6:53:56 PM, Sunday, May 6th, 2018

| SERVICE | OPERATION | START TIME | SUB-TRACE SUMMARY | SUB-TRACE DURATION |
|---|---|---|---|---|
| rainbow | router.update_span_group_statistics | 18:53:56 | 7 spans | 1460μs |
| rainbow | router.update_span_group_statistics | 18:53:56 | 7 spans | 1610μs |
| rainbow | router.update_span_group_statistics | 18:53:56 | 7 spans | 1260μs |

LIGHTSTEP

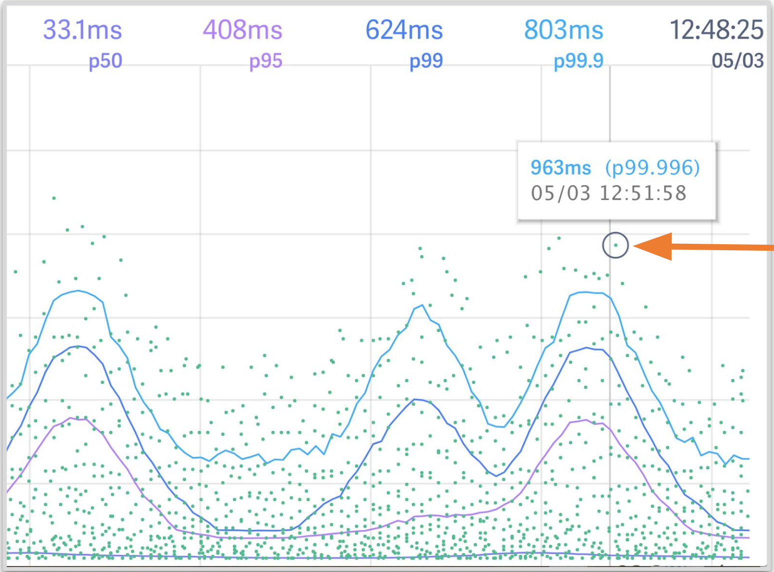
## Ideal Refinement: **Real-time**

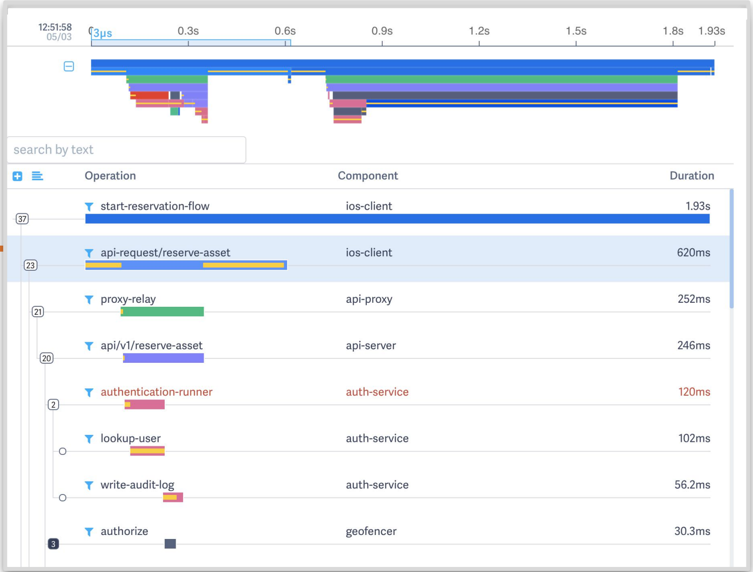
Must be able to **test and eliminate hypotheses** quickly

- Actual data must be ≤10s fresh
- UI / API latency must be very low

LIGHTSTEP

# Ideal Refinement: **Global**

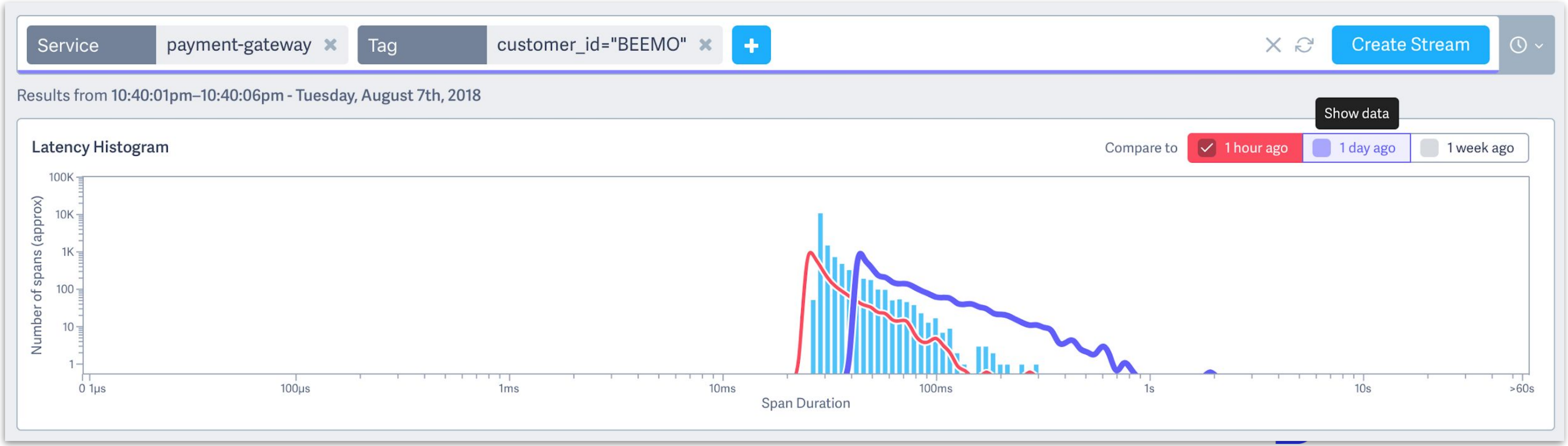




| Operation | Component | Duration |
|---|---|---|
| start-reservation-flow | ios-client | 1.93s |
| api-request/reserve-asset | ios-client | 620ms |
| proxy-relay | api-proxy | 252ms |
| api/v1/reserve-asset | api-server | 246ms |
| authentication-runner | auth-service | 120ms |
| lookup-user | auth-service | 102ms |
| write-audit-log | auth-service | 56.2ms |
| authorize | geofencer | 30.3ms |

# Ideal Refinement: **Context-Rich**

We can't expect humans to **know what's normal**



Results from 10:40:01pm–10:40:06pm - Tuesday, August 7th, 2018

Latency Histogram

## Thank you / Q&A