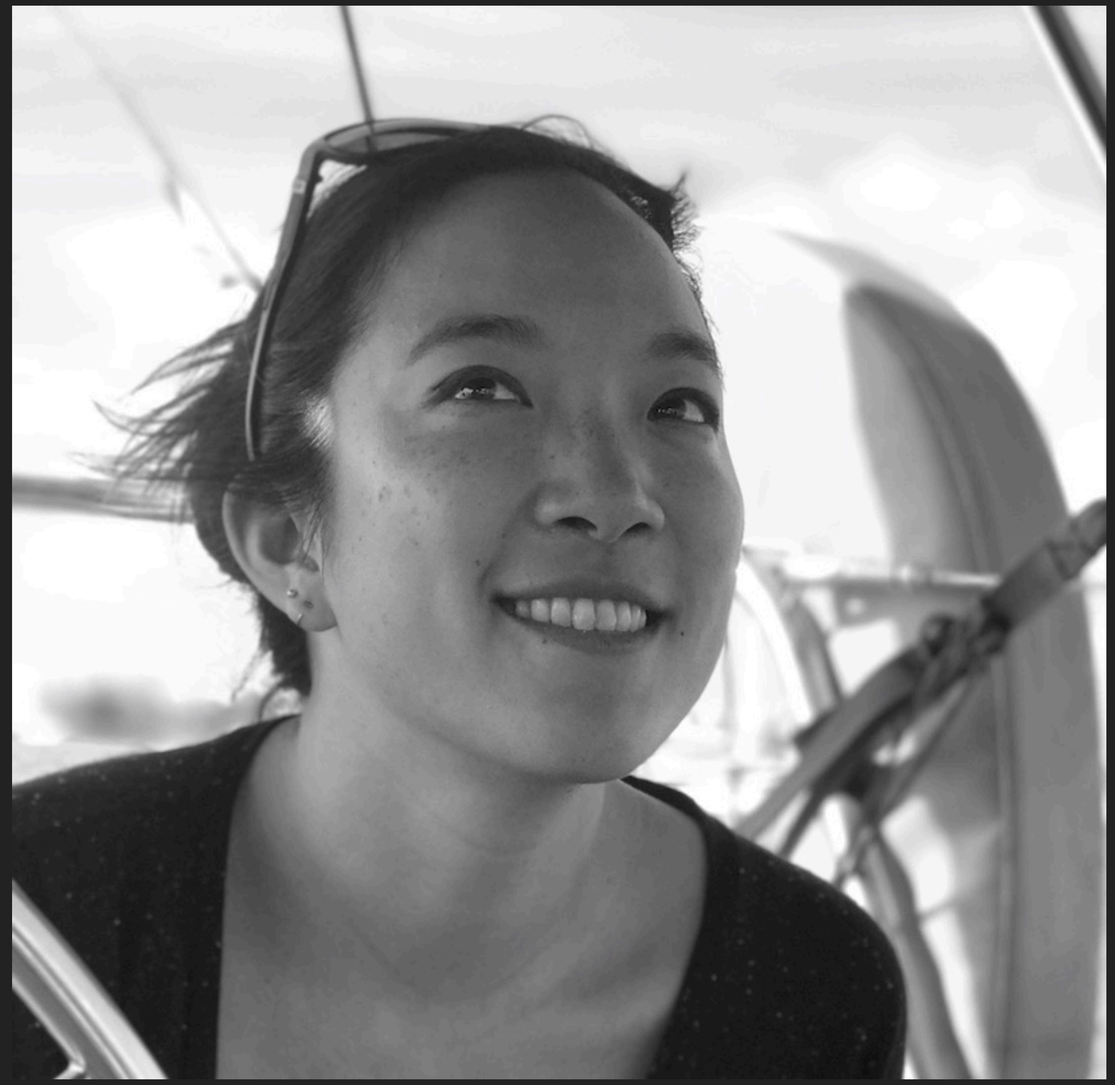
# observability for developers

## How to Get from Here to There

@cyen
@honeycombio

Christine

# DEV



# DEV

WRITE → TEST → COMMIT → WRITE → TEST → COMMIT
→ WRITE → TEST → COMMIT → WRITE → TEST → COMMIT
→ WRITE → TEST → COMMIT → WRITE → TEST → COMMIT
→ WRITE → TEST → COMMIT → WRITE → TEST → COMMIT

# DEV OPS

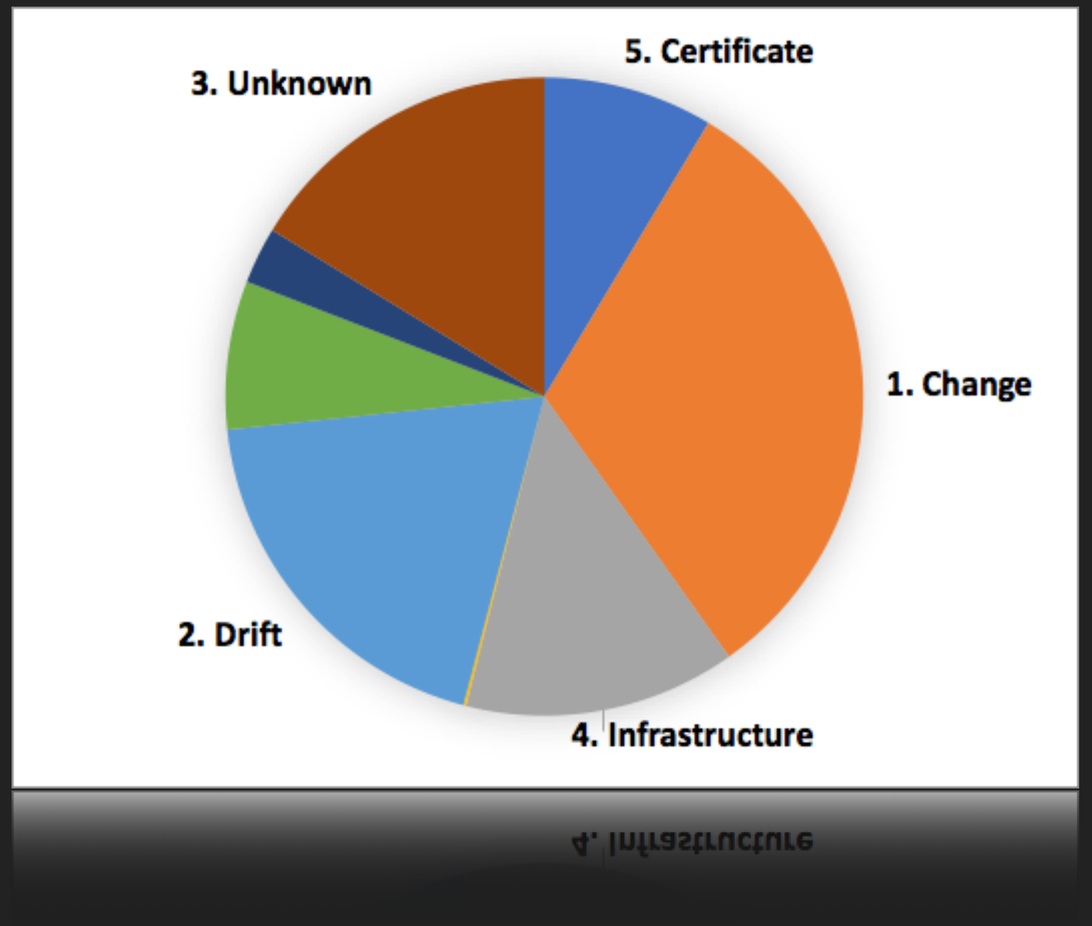
WRITE → TEST → COMMIT → RELEASE

# DEV  OPS

"Works on my machine"

"The only good diff is a red diff"

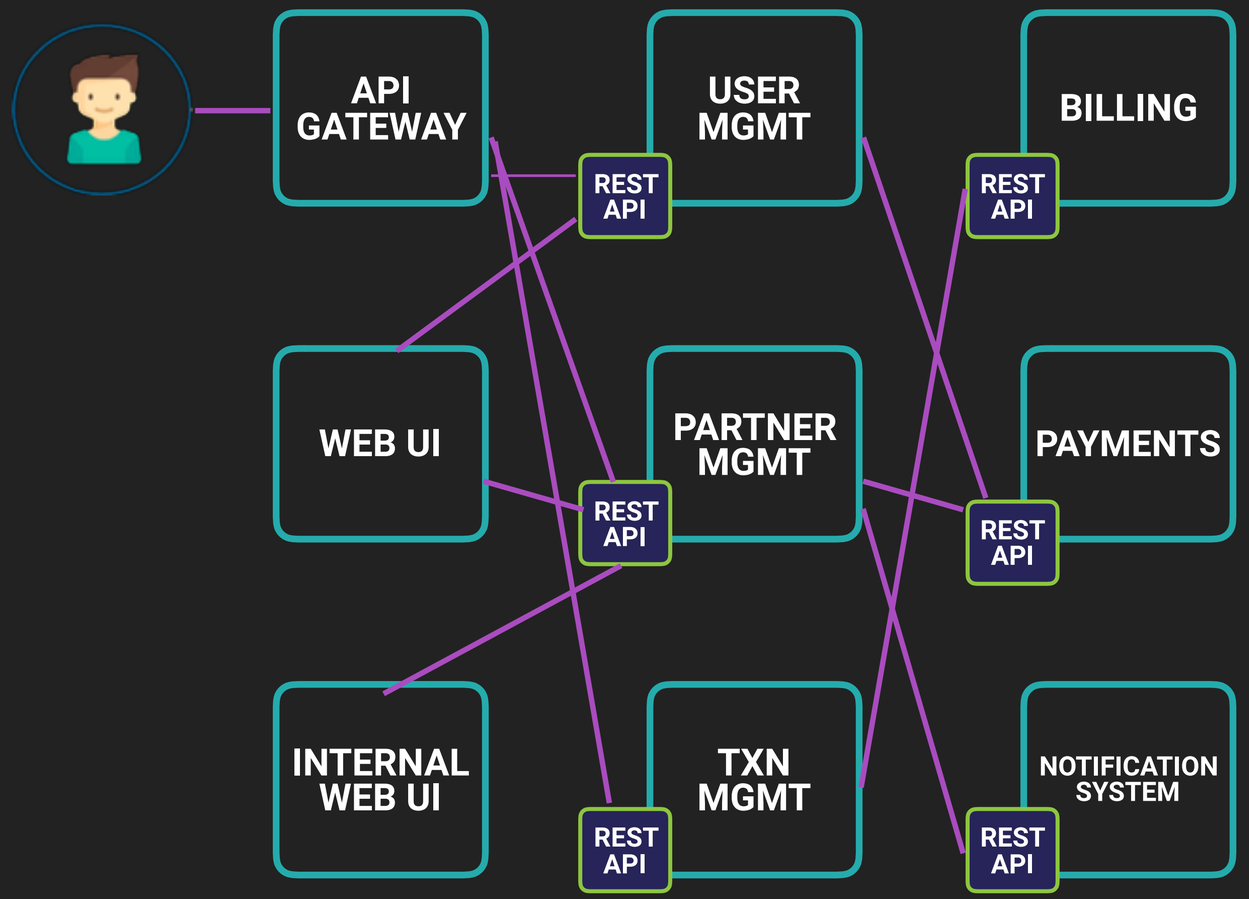
"Observation 1: Change is the most common trigger"




*—Subbu Allamaraju, Expedia, Feb 2019*
*https://m.subbu.org/incidents-trends-from-the-trenches-e2f8497d52ed*


APP

THEN


API GATEWAY
USER MGMT
BILLING
REST API
REST API
WEB UI
PARTNER MGMT
PAYMENTS
REST API
REST API
INTERNAL WEB UI
TXN MGMT
NOTIFICATION SYSTEM
REST API
REST API

NOW

# DEV  OPS

"Works on my machine"

"The only good diff is a red diff"

# DEV  OPS

**THE FIRST WAVE:** getting ops folks to code

**THE SECOND WAVE:** teaching devs to own code **in production**

# The Software **DEV** Process

- Design documents
- Architecture review
- Test-driven development
- Integration tests
- Code review
- Continuous integration
- Continuous deployment
- 🎉🥂🍾🎊
- Observe our code in production

# monitoring

The system as black box magic. Thresholds, alerts, system signals like CPU and memory.

Checking and rechecking for known bad behaviors.

# observability

The system as a living, adaptable thing. A culture of instrumentation and metadata rather than strictly-defined counters.

Being able to tease out previously-unknown bad behaviors and outliers.

# observability

a.k.a. understanding the behavior of
a system based on knowledge of its
external outputs.

a.k.a. "what is my software doing, and why
is it behaving that way?"

# DEV  OPS

"Works on my machine"

"The only good diff is a red diff"

**"How is it working for the user?"**

# What Does Observability-Driven Development

... look like?

# DEBUG PRODUCTION SYSTEMS

# DEBUG

- Locally: log lines, printfs, debuggers attached to our IDEs
- In production: we only have the data we captured when it happened
  - Make it **as easy as possible** to add new data as needed

# DEBUG

"My data isn’t showing up in Honeycomb!"

```
+ event_time_delta_sec
```

# Untitled Query

Add to Board | Share | Last 10 minutes

Datasets / shepherd

Add a description for this query

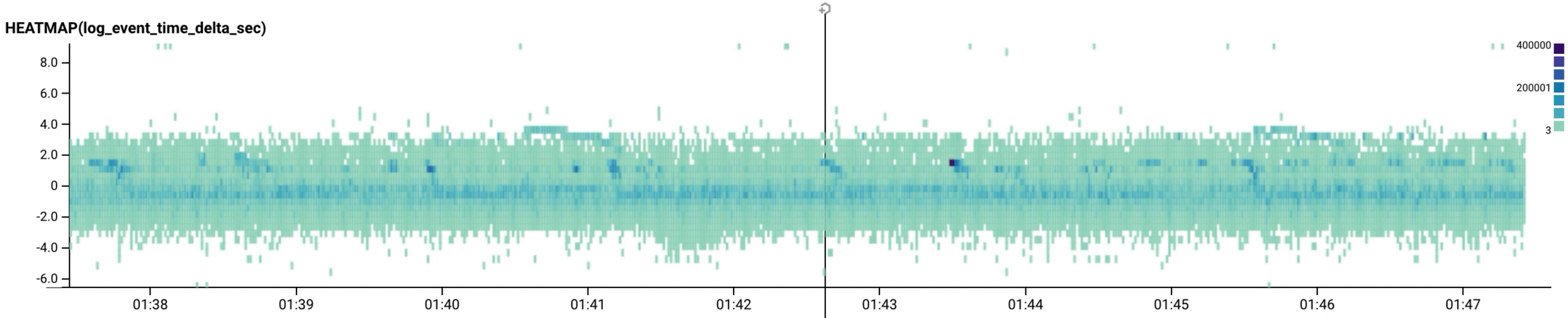
| VISUALIZE | WHERE | GROUP BY | ORDER BY | LIMIT |
|---|---|---|---|---|
| HEATMAP(log_event_time_delta_s<br>P95(app.content_length) | None; include all rows | None; combine all rows | None | None |

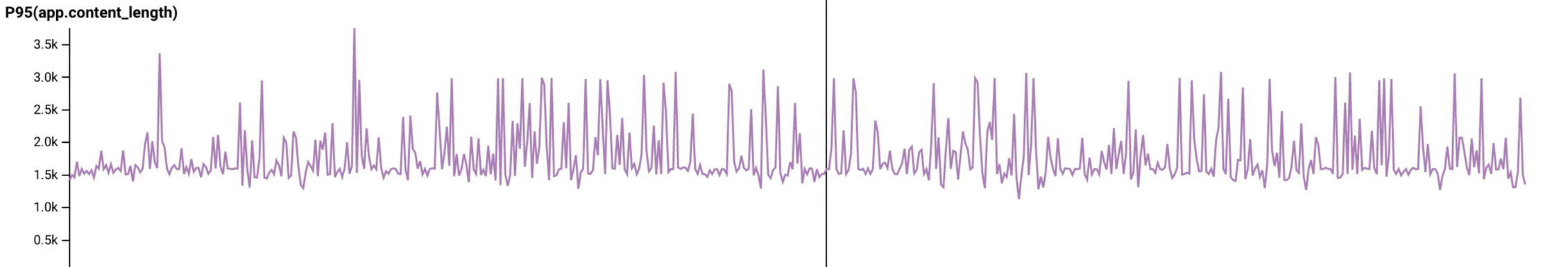
Run Query

Run a few seconds ago

Jun 12 2019, 1:37 AM – Jun 12 2019, 1:47 AM

Results | BubbleUp New | Traces | Raw Data | Graph Settings



# IMPROVE IN PROD

- "Test in Prod"...
  doesn't mean **only** testing in prod
- Testing: for **known knowns**
  Monitoring: for **known unknowns**
  Observability: for **unknown unknowns**
  —Jez Humble

IMPROVE

# FEATURE FLAGS 💞

IMPROVE

BREAK DOWN
None; combine all rows

CALCULATE
COUNT
SUM(count)
AVG(cum_write_latency_msec)
AVG(cum_write_latency_msec_per_

FILTER
None; include all rows

ORDER
COUNT desc

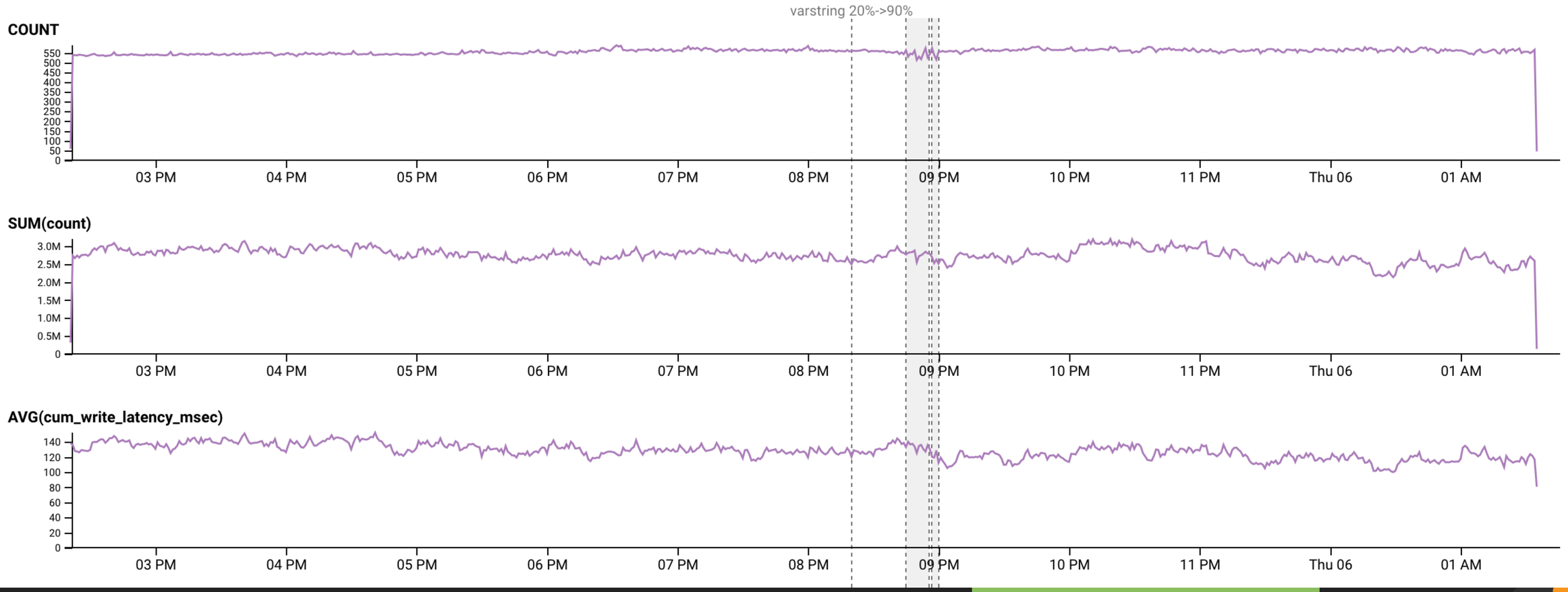
LIMIT
None

Run Query

Run 2 years ago

Jul 5 2017, 2:21 PM – Jul 6 2017, 1:35 AM

Results | BubbleUp New | Traces | Raw Data | Graph Settings



BREAK DOWN
flags.varstring

CALCULATE PER GROUP
COUNT
unt)
m_write_latency_msec)
m_write_latency_msec_per_

FILTER
None; include all rows

ORDER
COUNT desc

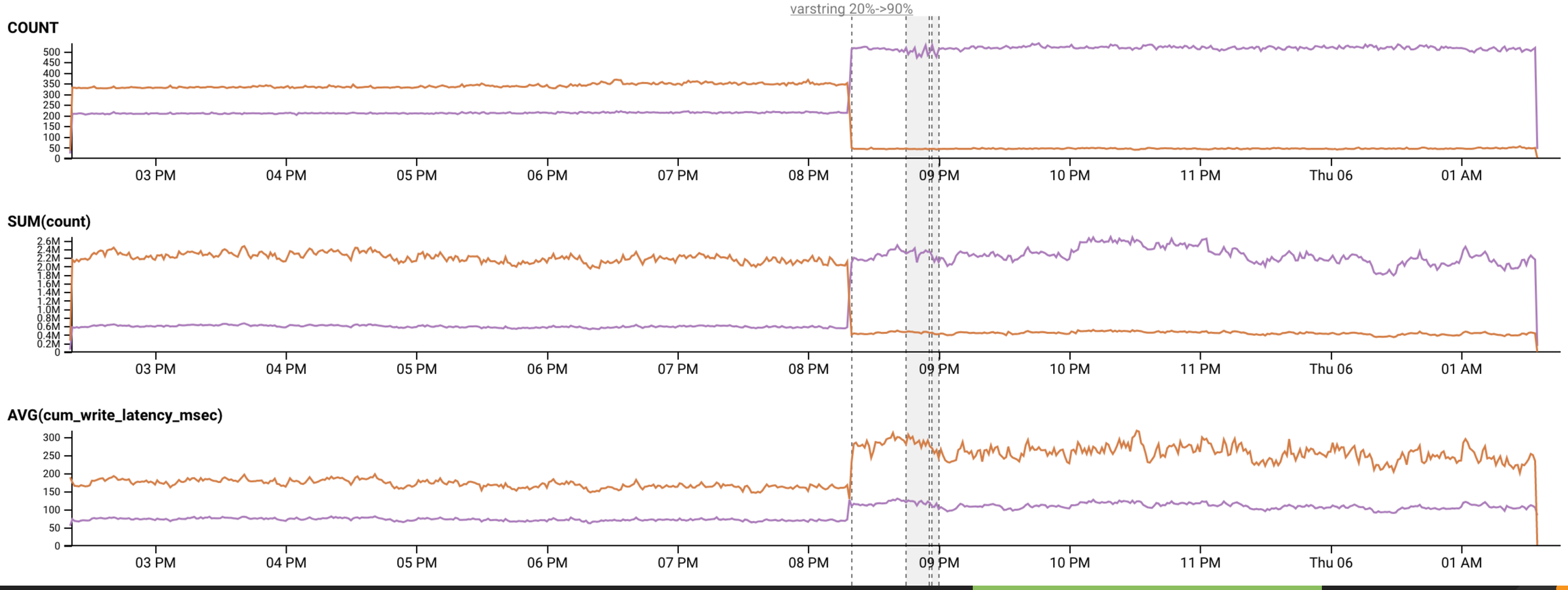
LIMIT
None

Run Query
Run 2 years ago

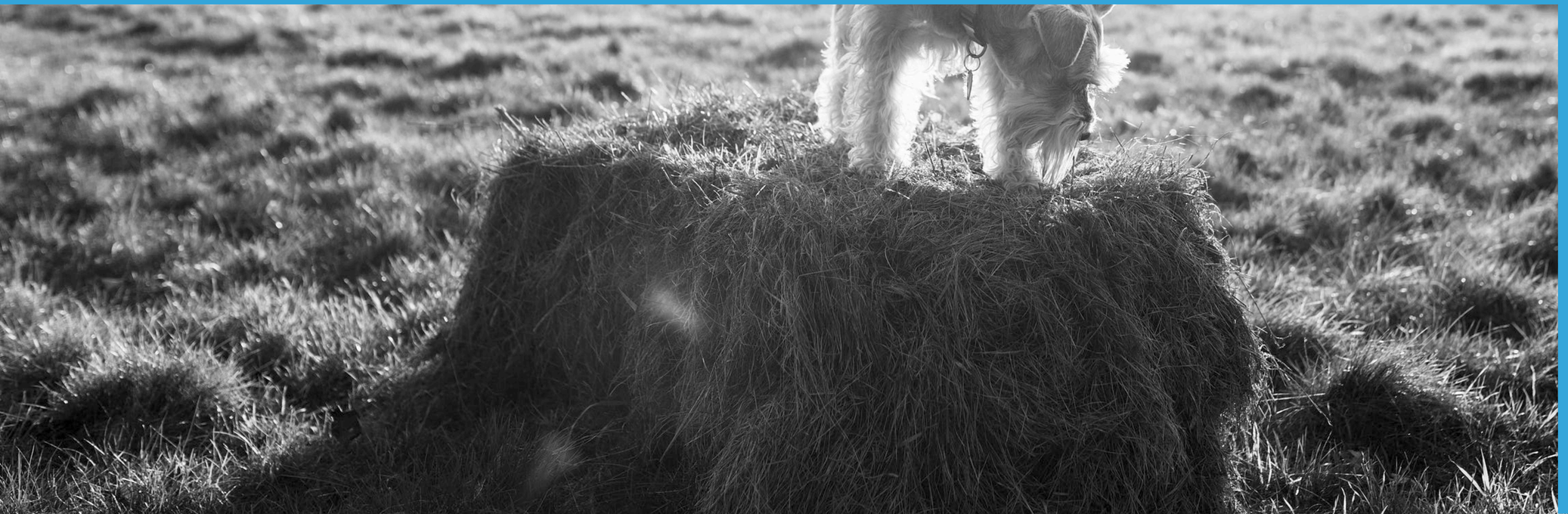
BREAK DOWN
flags.varstring

Jul 5 2017, 2:21 PM – Jul 6 2017, 1:35 AM

Results | BubbleUp New | Traces | Raw Data | Graph Settings

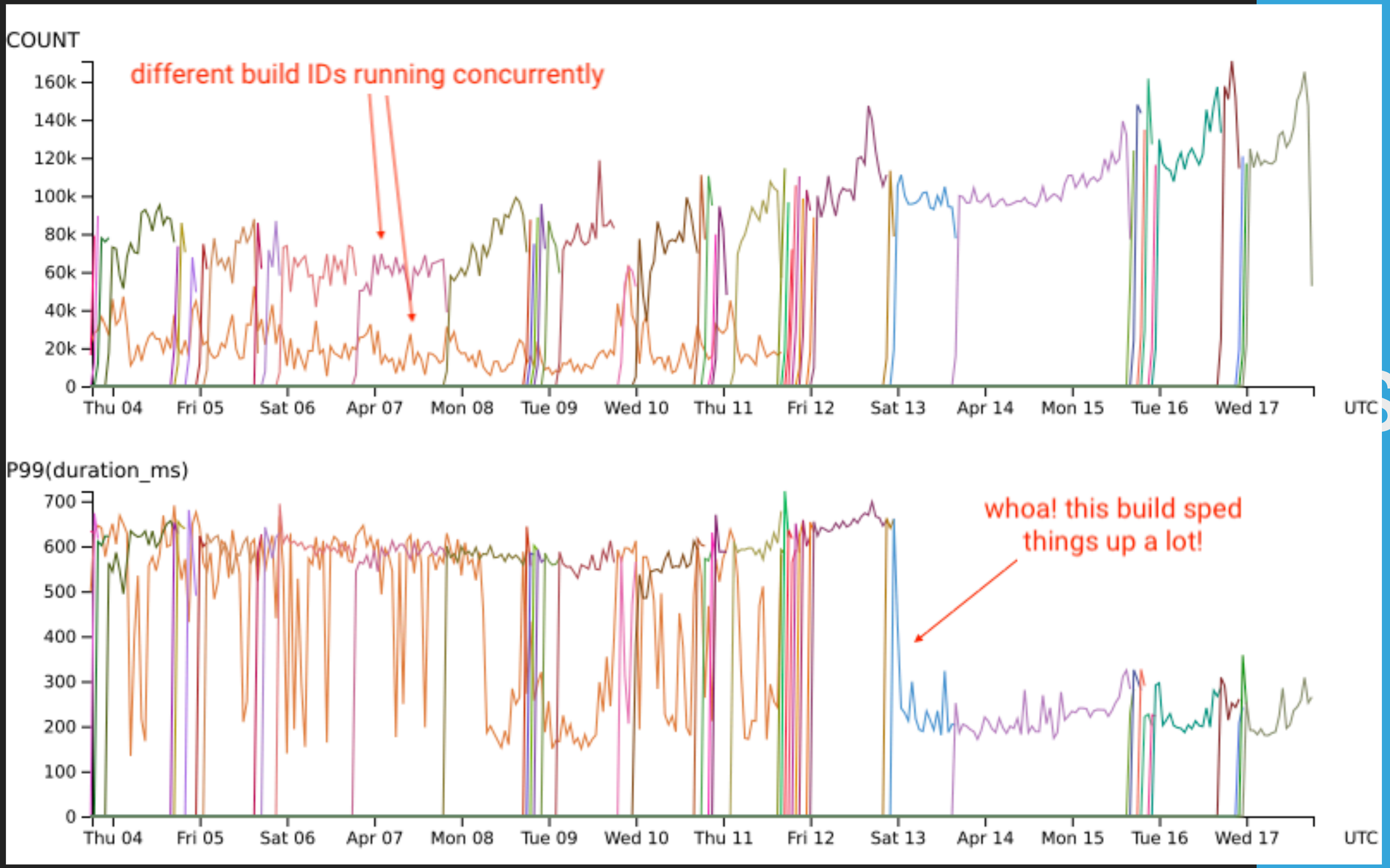


# IS IT STILL WORKING? LET’S OBSERVE



# OBSERVE

- Watch to make sure reality lines up with expectations
- … in the terms that **we** understand intimately


COUNT
different build IDs running concurrently
160k
140k
120k
100k
80k
60k
40k
20k
0
Thu 04
Fri 05
Sat 06
Apr 07
Mon 08
Tue 09
Wed 10
Thu 11
Fri 12
Sat 13
Apr 14
Mon 15
Tue 16
Wed 17
UTC
P99(duration_ms)
700
600
500
400
300
200
100
0
whoa! this build sped things up a lot!
Thu 04
Fri 05
Sat 06
Apr 07
Mon 08
Tue 09
Wed 10
Thu 11
Fri 12
Sat 13
Apr 14
Mon 15
Tue 16
Wed 17
UTC

SERVE

- Instrumentation (Getting Data In)
  - Best Practices
  - Taking the First Few Steps
  - Migrating from Unstructured Text Logs
- Stop Searching, Start Analyzing
- Tracing as a New Frontier

# BEST PRACTICES FOR INSTRUMENTATION

- Capture contextual, structured data

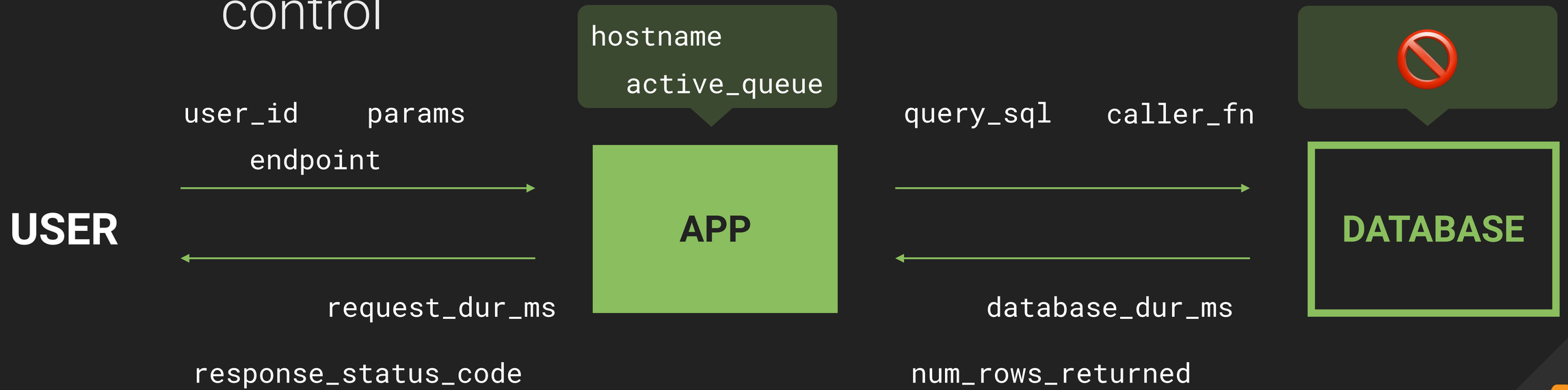
```
{
  Timestamp: "2018-03-20T00:47:25.339Z",
  content_length: 172,
  database_dur_ms: 15.79283,
  endpoint: "/posts/15",
  method: "PUT",
  request_dur_ms: 72.446625,
  render_dur_ms: 25.31729,
  service_name: "api",
  user_token: "2e6cfd4"
}
```

# BEST PRACTICES FOR INSTRUMENTATION

- Capture contextual, structured data
- Common set of nouns and consistent naming

# BEST PRACTICES FOR INSTRUMENTATION

- Capture contextual, structured data
- Common set of nouns and consistent naming
- Instrument from the perspective of what you **can** control



# TAKING THE FIRST FEW STEPS

- Describe your basic "unit of work" and identify where it "enters" the system

# TAKING THE FIRST FEW STEPS

- Describe your basic "unit of work" and identify where it "enters" the system
- Identify metadata to help you isolate unexpected behavior in your business logic

| Your Infra | Your Deploy | Your Business | Your Execution |
|---|---|---|---|
| - hostname<br>- machine type | - version / build<br>- feature flags | - customer<br>- shopping cart | - payload characteristics<br>- timers |

# TAKING THE FIRST FEW STEPS

- Describe your basic "unit of work" and identify where it "enters" the system
- Identify metadata to help you isolate unexpected behavior in your business logic
- Experiment! Add temporary fields when needed to validate hypotheses

# TAKING THE FIRST FEW STEPS

- Describe your basic "unit of work" and identify where it "enters" the system
- Identify metadata to help you isolate unexpected behavior in your business logic
- Experiment! Add temporary fields when needed to validate hypotheses
- Prune stale fields (if necessary)

# MIGRATING FROM UNSTRUCTURED TEXT LOGS

```
2019-01-25T01:30:23.743Z Enqueued task
2019-01-25T01:30:24.120Z Task processed, returning 42 entries
2019-01-25T01:30:24.212Z Task complete (email sent to foobar@example.com)
2019-01-25T01:30:26.014Z Enqueued task
2019-01-25T01:30:26.214Z Enqueued task
2019-01-25T01:30:24.120Z Task errored: unknown constant ::Fixnum
2019-01-25T01:30:29.953Z Task timed out after 6.01 seconds
2019-01-25T01:30:32.762Z Enqueued task
2019-01-25T01:30:32.791Z Enqueued task
2019-01-25T01:30:32.993Z Task processed, returning 7 entries
2019-01-25T01:30:33.132Z Task complete (email not found, noop)
2019-01-25T01:30:34.243Z Task processed, returning 0 entries
2019-01-25T01:30:34.243Z Task complete, (email sent to bazqux@example.com)
```

# MIGRATING FROM UNSTRUCTURED TEXT LOGS

- Identify entities that are relevant to your business logic (and include them in your logs!)

```
2019-01-25T01:30:29.953Z Task timed out after 6.01 seconds task_id=72 type=process
```

# MIGRATING FROM UNSTRUCTURED TEXT LOGS

- Identify entities that are relevant to your business logic (and include them in your logs!)
- Start introducing structure into your logs

```
2019-01-25T01:30:29.953Z Task timed out after 6.01 seconds task_id=72 type=process
```

```
Timestamp=2019-01-25T01:30:29.953Z
 message=Task timed out after 6.01 seconds
 task_id=72
 type=process
```

# MIGRATING FROM UNSTRUCTURED TEXT LOGS

- Identify entities that are relevant to your business logic (and include them in your logs!)
- Start introducing structure into your logs
- Build up **context** instead of outputting disjoint lines

```
2019-01-25T01:30:23.743Z Enqueued task task_id=72 type=enqueue target=email
2019-01-25T01:30:29.953Z Task timed out after 6.01 seconds task_id=72 type=process
```

```
Timestamp=2019-01-25T01:30:29.953Z
 message=Task timed out after 6.01 seconds
 task_id=72
 target=email
 queue_dur_ms=200
 timeout_dur_ms=6010
```

# STOP SEARCHING, START ANALYZING

- Logs were conceived to store and find history, not for analytics

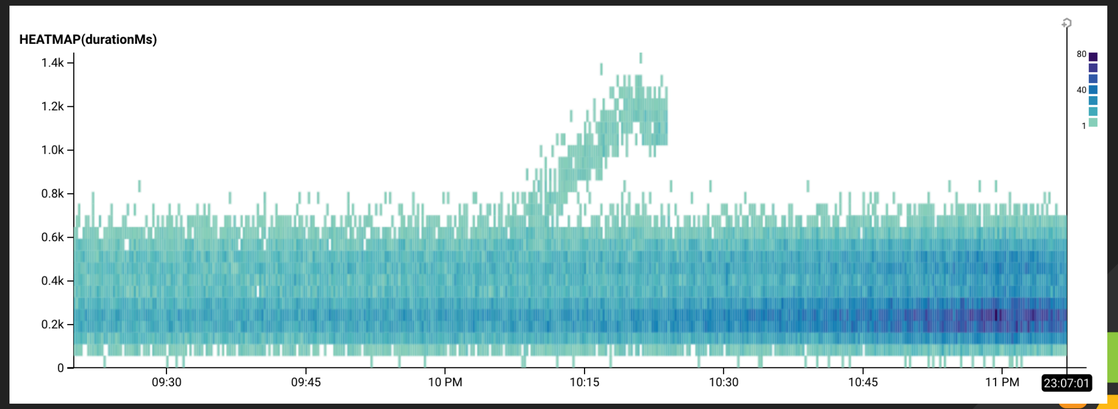
```
2019-01-25T01:30:23.743Z Enqueued task
2019-01-25T01:30:24.120Z Task processed, returning 42 entries
2019-01-25T01:30:24.212Z Task complete (email sent to foobar@example.com)
2019-01-25T01:30:26.014Z Enqueued task
2019-01-25T01:30:26.214Z Enqueued task
2019-01-25T01:30:24.120Z Task errored: unknown constant ::Fixnum
2019-01-25T01:30:29.953Z Task timed out after 6.01 seconds
2019-01-25T01:30:32.762Z Enqueued task
2019-01-25T01:30:34.243Z Task processed, returning 0 entries
2019-01-25T01:30:34.243Z Task complete, (email sent to bazqux@example.com)
```

# STOP SEARCHING, START ANALYZING

- Logs were conceived to store and find history, not for analytics
- Logs **are no longer human-scale** — they are machine-scale

# STOP SEARCHING, START ANALYZING

- Logs were conceived to store and find history, not for analytics
- Logs **are no longer human-scale** — they are machine-scale
- Visualizations are necessary to identify an outlier as a trend or an anomaly



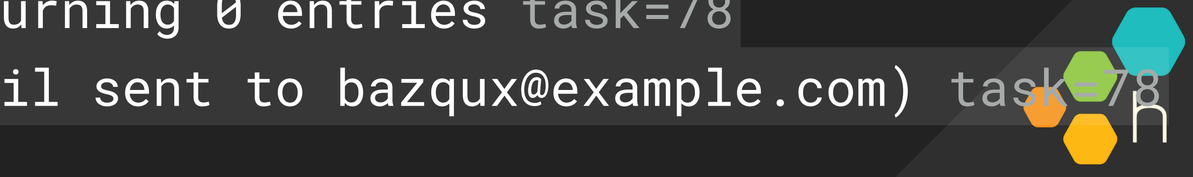
# TRACING AS A NEW FRONTIER

- Tracing: not just for concurrent or distributed systems

# TRACING AS A NEW FRONTIER

- Tracing: not just for concurrent or distributed systems

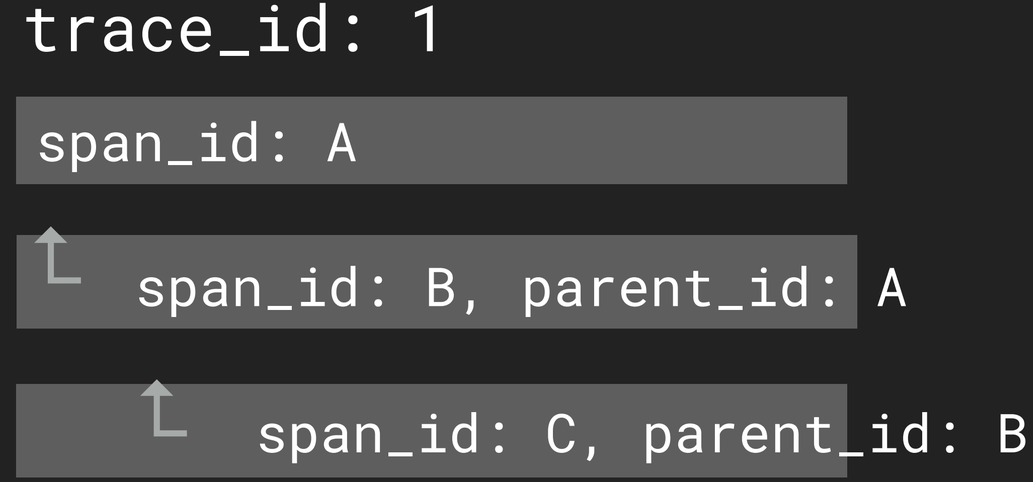
```
2019-01-25T01:30:23.743Z Enqueued task task=72
2019-01-25T01:30:24.120Z Enqueued task task=74
2019-01-25T01:30:24.212Z Task processed, returning 42 entries task=74
2019-01-25T01:30:26.014Z Task complete (email sent to foobar@example.com) task=74
2019-01-25T01:30:26.214Z Enqueued task task=77
2019-01-25T01:30:24.120Z Task errored: unknown constant ::Fixnum task=77
2019-01-25T01:30:29.953Z Task timed out after 6.01 seconds task=72
2019-01-25T01:30:32.762Z Enqueued task task=78
2019-01-25T01:30:34.243Z Task processed, returning 0 entries task=78
2019-01-25T01:30:34.243Z Task complete, (email sent to bazqux@example.com) task=78
```



# TRACING AS A NEW FRONTIER

- Tracing: not just for concurrent or distributed systems
- A series of related log lines can, in fact, share a **lot** in common with a trace

```
service_name    trace_id
name            span_id
duration_ms     parent_id
```



# Untitled Query

Datasets / basset

Add a description for this query

Add to Board | Share | ••• | 1/24 12:55pm to 1/25 12:55am

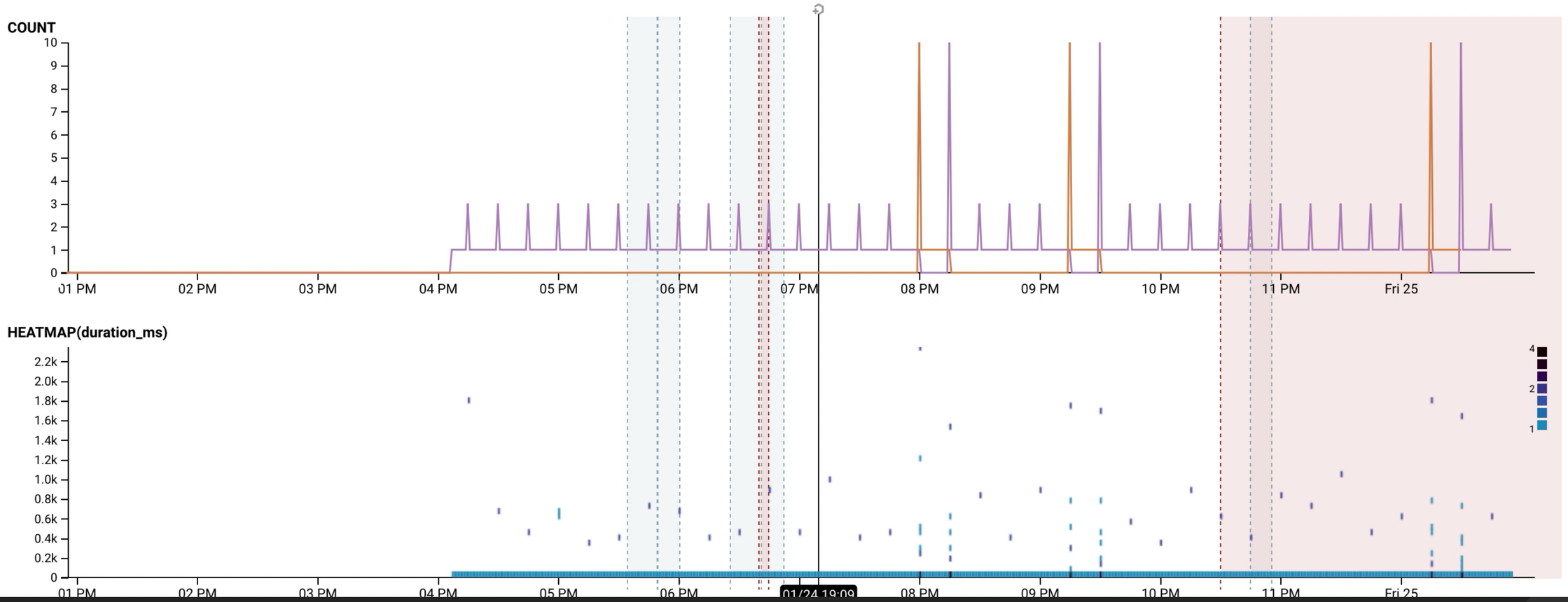
| BREAK DOWN | CALCULATE PER GROUP | FILTER | ORDER | LIMIT |
|---|---|---|---|---|
| app.trigger.status | COUNT<br>HEATMAP(duration_ms) | app.trigger.id = zeiynqf9UWm | COUNT desc | None |

Run Query

Run a few seconds ago

Jan 24 2019, 12:55 PM – Jan 25 2019, 12:55 AM

Results | BubbleUp New | Traces | Raw Data | Graph Settings



# TRACING AS A NEW FRONTIER

- Tracing: not just for concurrent or distributed systems
- A series of related log lines can, in fact, share a **lot** in common with a trace
- Tracing will be commonplace in 2019 [0]

0: https://monitoring.love/articles/2019-predictions/

# TRACING AS A NEW FRONTIER

- Tracing: not just for concurrent or distributed systems
- A series of related log lines can, in fact, share a **lot** in common with a trace
- Tracing will be commonplace in 2019
- Aggregate analysis of traces is still key

# DEV

# DEV OPS

# DEV OPS

DEV

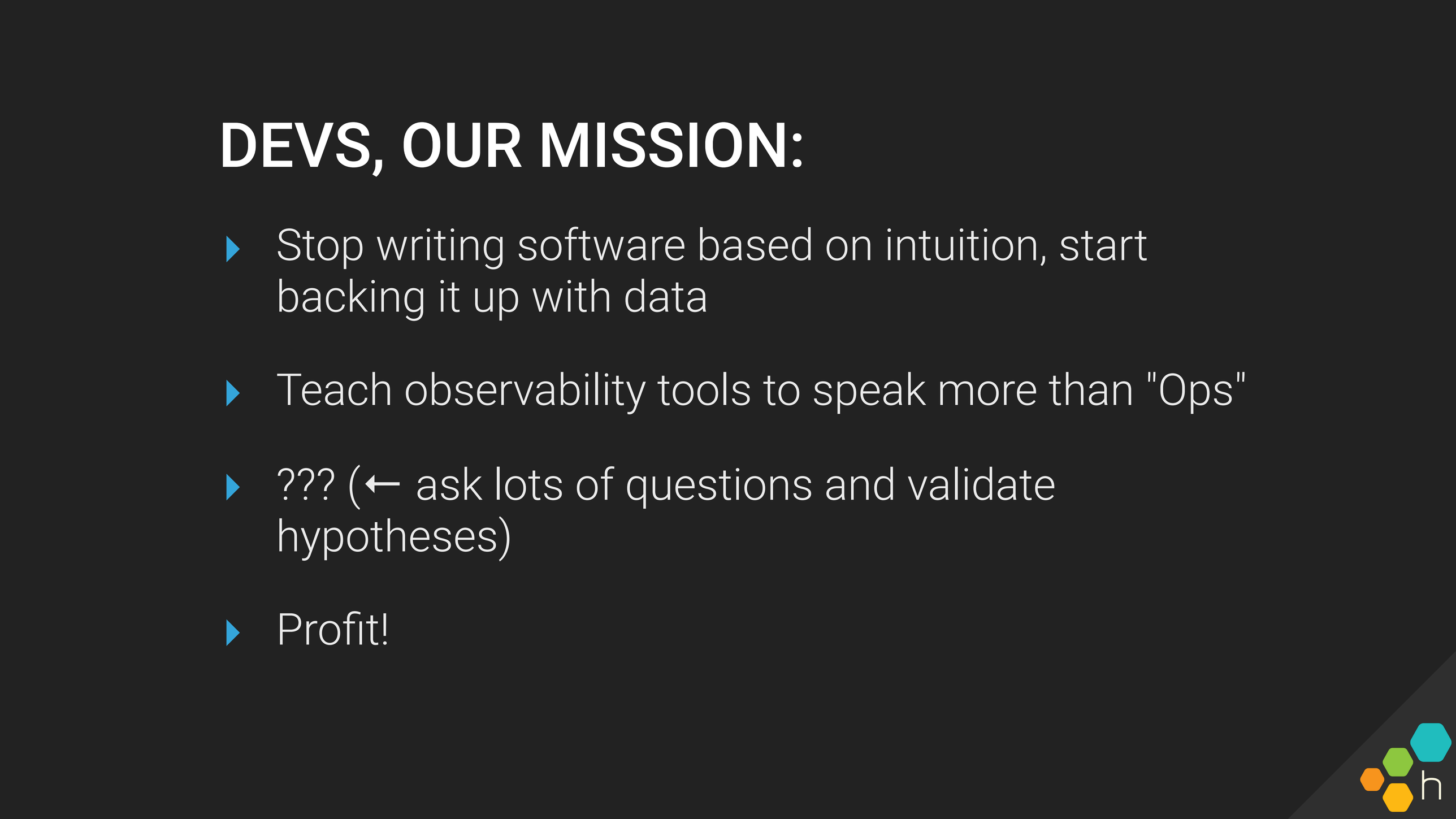
OPS

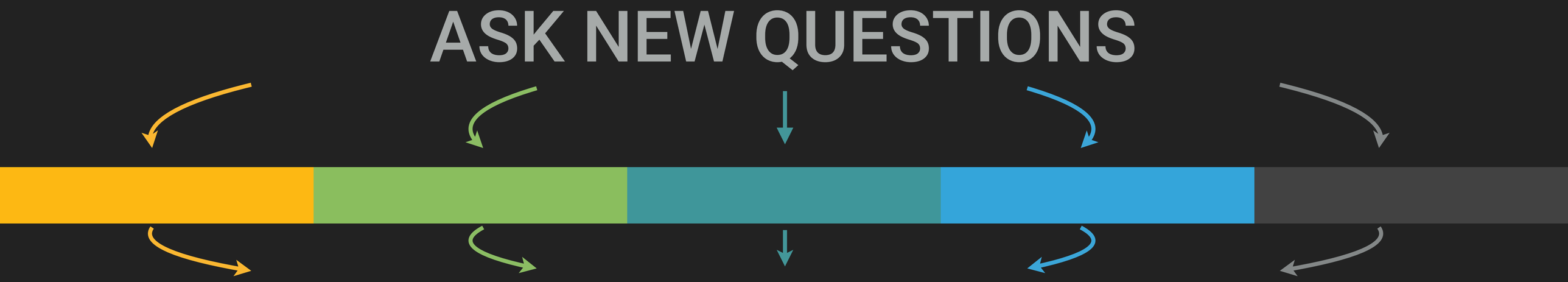
DEV

OPS

# DEVS, OUR MISSION:

- Stop writing software based on intuition, start backing it up with data
- Teach observability tools to speak more than "Ops"
- ??? (← ask lots of questions and validate hypotheses)
- Profit!






# thanks!

@cyen
@honeycombio