CIRCONUS

Put Some SRE in Your Shipped Software

Hard-won lessons from the world of SRE

Theo Schlossnagle

CEO, Circonus

@postwait

CIRCONUS

The nature of the problem:

Software Sucks.

Once you've run software at scale, you have a deep understanding of how it is all tied together with **loose string** and **hope**.

We spend massive effort to operationalize our stacks.

The burning question:

Why Ship Software?

The trends are there.
You can choose to see them or not.

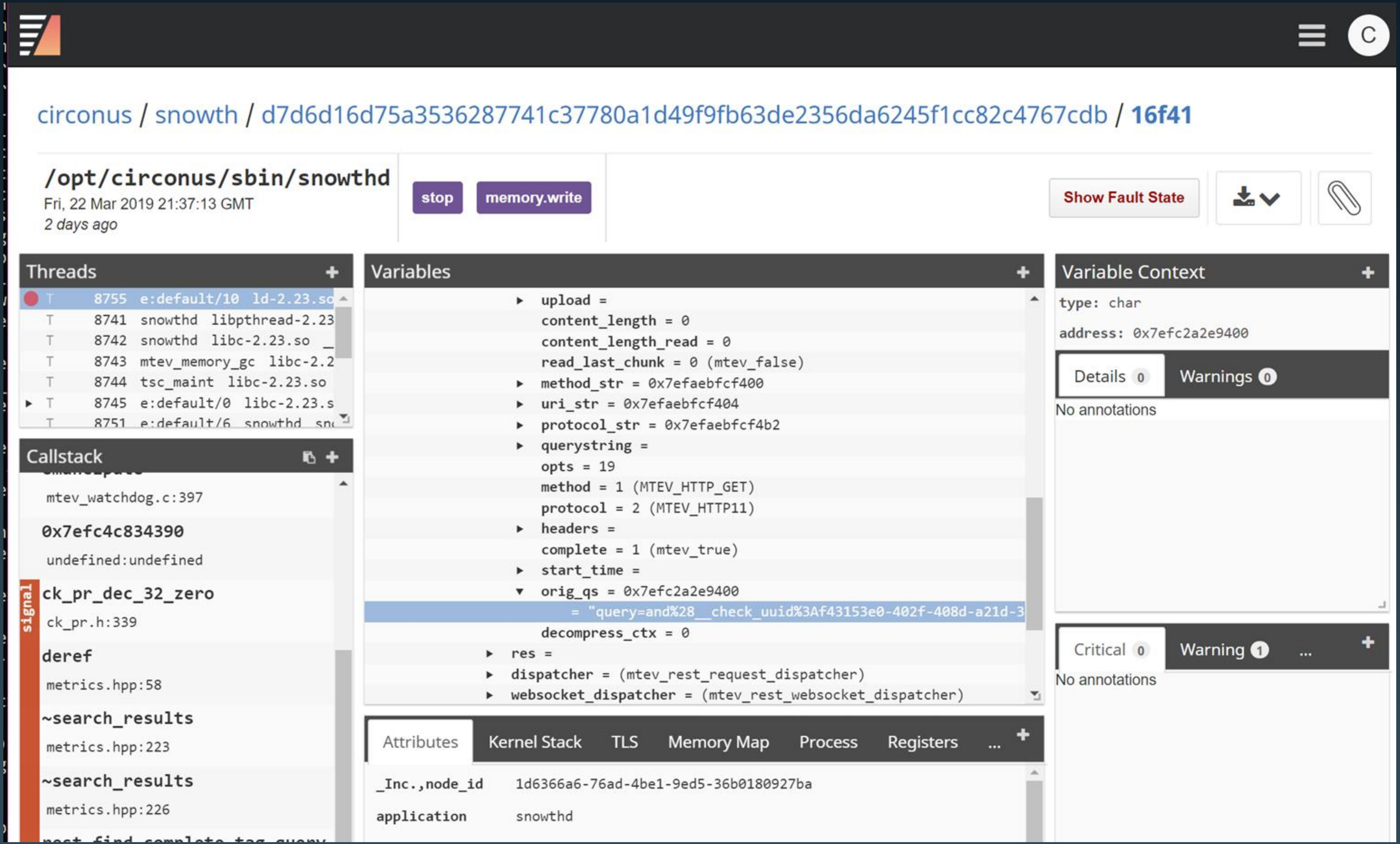
Rules.

1. Crash landings should be both fast and controlled.
2. Post-mortems are fundamental.
3. Use circuit breakers.
4. Behavior is complex. Understand it.
5. Have a failure budget.
6. Instrumentation & observability have no equals.

Build upon the right layers

Crash Analysis

If you don't know why it failed,
you dont' know anything at all.


circonus / snowth / d7d6d16d75a3536287741c37780a1d49f9fb63de2356da6245f1cc82c4767cdb / **16f41**

```
/opt/circonus/sbin/snowthd
```
Fri, 22 Mar 2019 21:37:13 GMT
2 days ago

Threads

```
● T  8755  e:default/10  ld-2.23.so
  T  8741  snowthd  libpthread-2.23
  T  8742  snowthd  libc-2.23.so  _
  T  8743  mtev_memory_gc  libc-2.2
  T  8744  tsc_maint  libc-2.23.so
▸ T  8745  e:default/0  libc-2.23.s
  T  8751  e:default/6  snowthd  sn
```

Callstack

```
mtev_watchdog.c:397
0x7efc4c834390
undefined:undefined
signal
ck_pr_dec_32_zero
ck_pr.h:339
deref
metrics.hpp:58
~search_results
metrics.hpp:223
~search_results
metrics.hpp:226
```

Variables

```
    ▸ upload =
      content_length = 0
      content_length_read = 0
      read_last_chunk = 0 (mtev_false)
    ▸ method_str = 0x7efaebfcf400
    ▸ uri_str = 0x7efaebfcf404
    ▸ protocol_str = 0x7efaebfcf4b2
    ▸ querystring =
      opts = 19
      method = 1 (MTEV_HTTP_GET)
      protocol = 2 (MTEV_HTTP11)
    ▸ headers =
      complete = 1 (mtev_true)
    ▸ start_time =
    ▾ orig_qs = 0x7efc2a2e9400
          = "query=and%28__check_uuid%3Af43153e0-402f-408d-a21d-3
      decompress_ctx = 0
  ▸ res =
  ▸ dispatcher = (mtev_rest_request_dispatcher)
  ▸ websocket_dispatcher = (mtev_rest_websocket_dispatcher)
```

```
_Inc.,node_id    1d6366a6-76ad-4be1-9ed5-36b0180927ba
application      snowthd
```

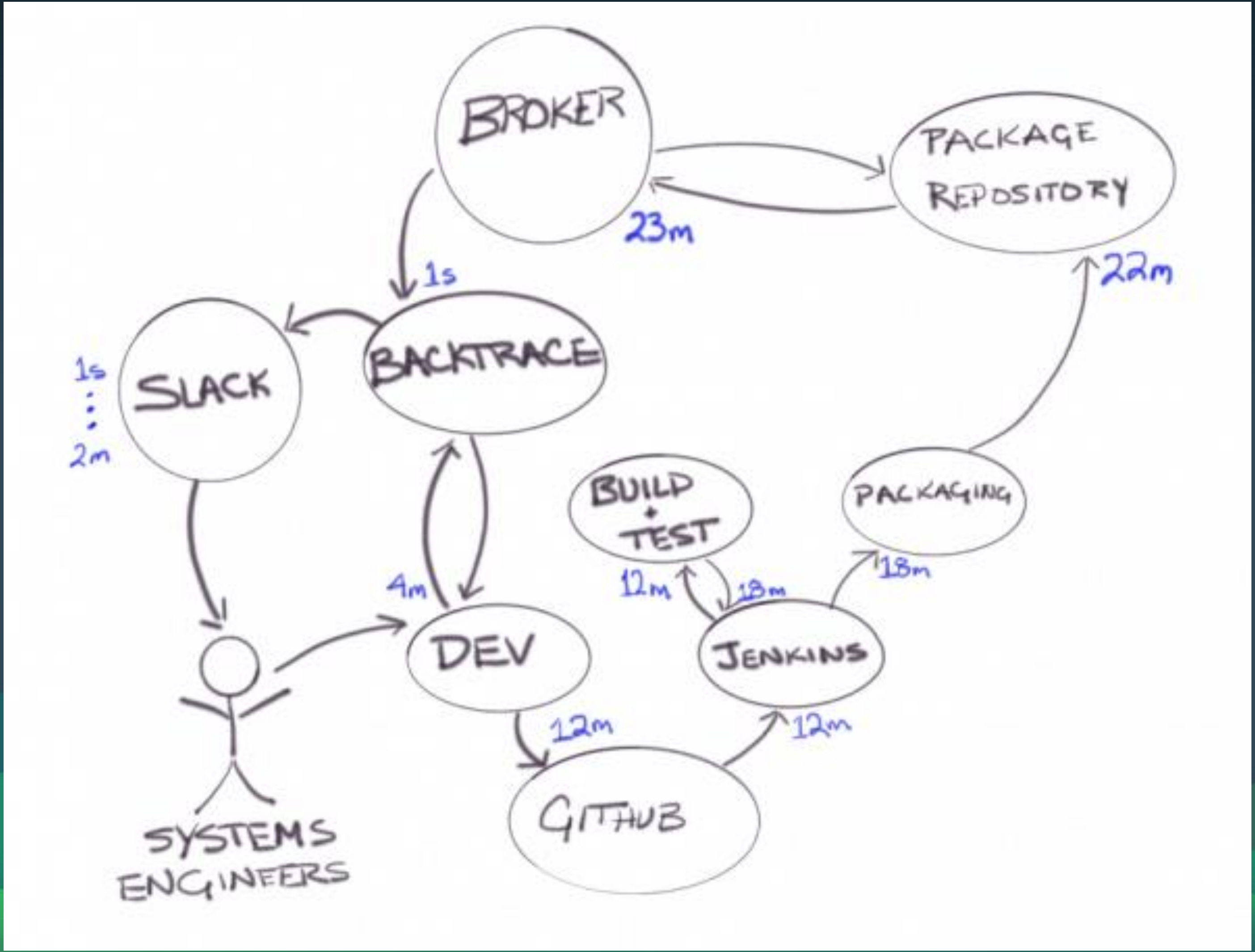
Variable Context

```
type: char
address: 0x7efc2a2e9400
```

No annotations

No annotations

CIRCONUS


BROKER
PACKAGE REPOSITORY
23m
22m
1s
BACKTRACE
SLACK
1s
⋮
2m
4m
BUILD + TEST
PACKAGING
12m
18m
18m
DEV
JENKINS
12m
12m
GITHUB
SYSTEMS ENGINEERS

When in doubt or even curious

Expose Telemetry

Ideally, any question you would ask of a production system can be done so nondisruptively.

CIRCONUS

libcircmetrics

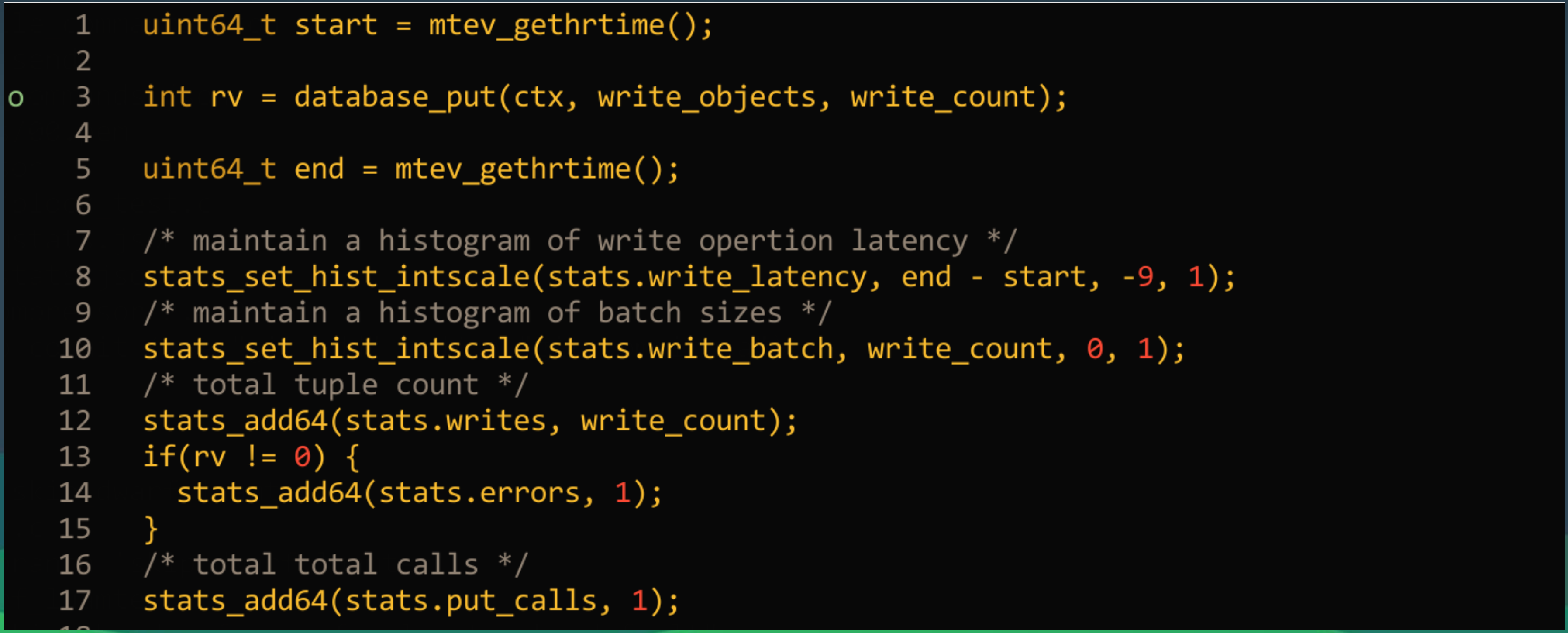
- C library; BSD license, fast, thread-safe, largely lockless.
    - Text metrics (version numbers, statuses, etc.)
    - Numeric gauges, counters (w/ CPU fanout)
    - Histograms (log-linear quantized) (9ns recording)
    - Simultaneous hierarchical (graphite-style) and tagged annotation support
    - JSON output

Code Sample

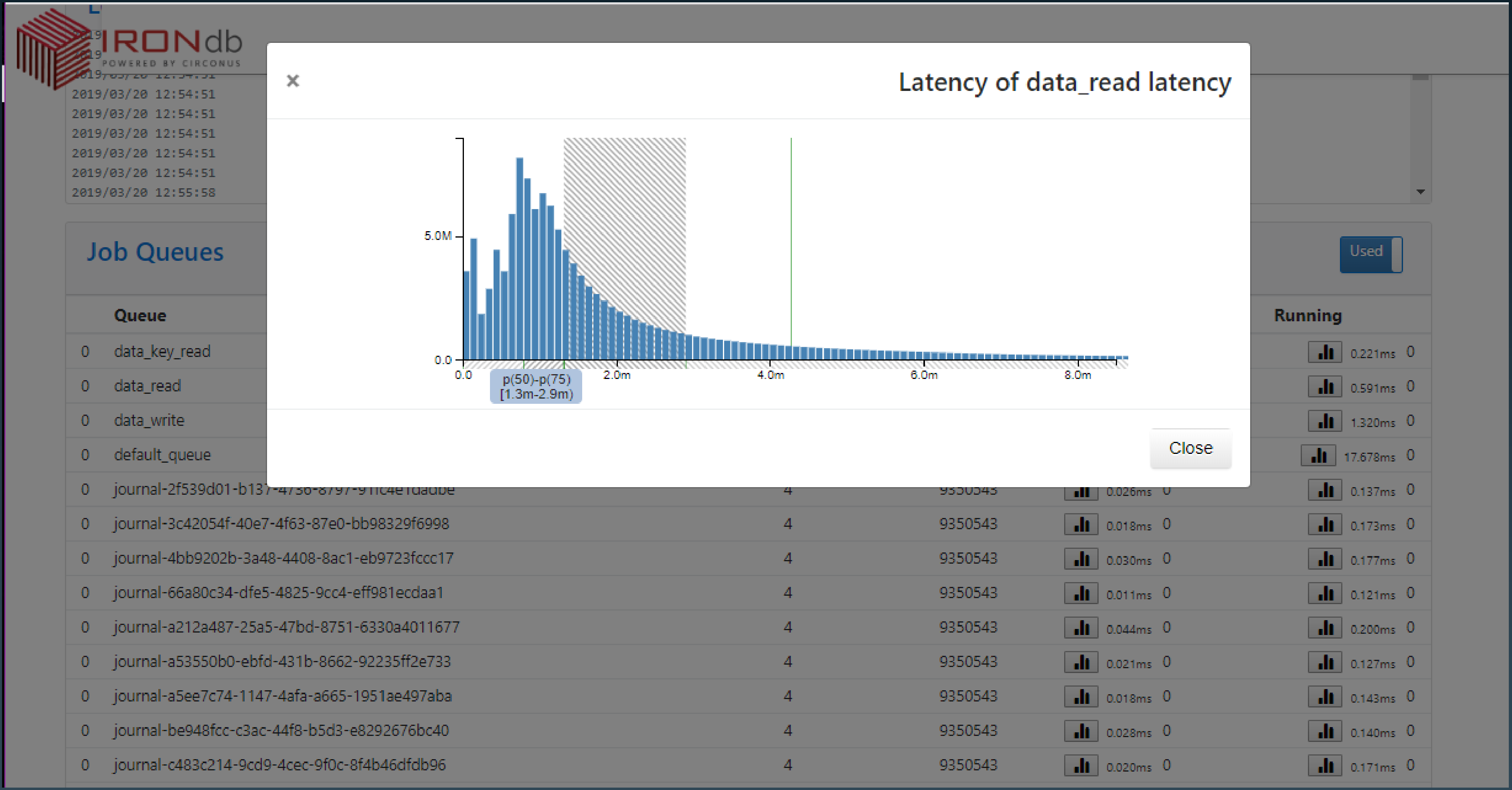
```
stats_recorder_t *rec = stats_recorder_alloc();
stats_ns_t *ns = stats_register_ns(rec, NULL, "db");
stats_handle_t *h;

ns = stats_register_ns(rec, ns, "raw");
stats_ns_add_tag(nomns, "db-type", "raw");
stats_ns_add_tag(nomns, "db-impl", "nom");

h = stats.put_calls = stats_register(nomns, "put.calls", STATS_TYPE_COUNTER);
stats_handle_tagged_name(h, "calls");
stats_handle_add_tag(h, "operation", "put");
stats_handle_units(h, STATS_UNITS_REQUESTS);

h = stats.get_calls = stats_register(nomns, "get.calls", STATS_TYPE_COUNTER);
stats_handle_tagged_name(h, "calls");
stats_handle_add_tag(h, "operation", "get");
stats_handle_units(h, STATS_UNITS_REQUESTS);

h = stats.writes = stats_register(nomns, "put.tuples", STATS_TYPE_COUNTER);
stats_handle_tagged_name(h, "count");
stats_handle_add_tag(h, "operation", "put");
stats_handle_units(h, STATS_UNITS_TUPLES);

h = stats.write_latency = stats_register(nomns, "put.latency", STATS_TYPE_HISTOGRAM);
stats_handle_tagged_name(h, "latency");
stats_handle_add_tag(h, "operation", "put");
stats_handle_units(h, STATS_UNITS_SECONDS);

h = stats.write_batch = stats_register(nomns, "put.batchsize", STATS_TYPE_HISTOGRAM);
stats_handle_tagged_name(h, "batchsize");
stats_handle_add_tag(h, "operation", "put");
stats_handle_units(h, STATS_UNITS_TUPLES);
```

CIRCONUS

Code Sample

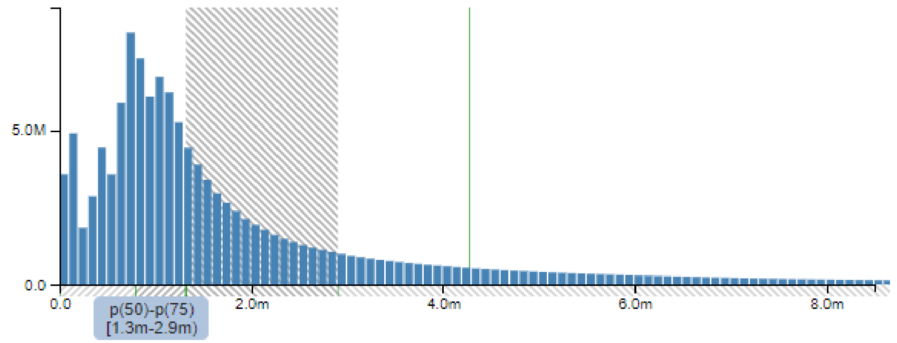


```c
uint64_t start = mtev_gethrtime();

int rv = database_put(ctx, write_objects, write_count);

uint64_t end = mtev_gethrtime();

/* maintain a histogram of write opertion latency */
stats_set_hist_intscale(stats.write_latency, end - start, -9, 1);
/* maintain a histogram of batch sizes */
stats_set_hist_intscale(stats.write_batch, write_count, 0, 1);
/* total tuple count */
stats_add64(stats.writes, write_count);
if(rv != 0) {
  stats_add64(stats.errors, 1);
}
/* total total calls */
stats_add64(stats.put_calls, 1);
```



```
2019/03/20 12:54:51
2019/03/20 12:54:51
2019/03/20 12:54:51
2019/03/20 12:54:51
2019/03/20 12:54:51
2019/03/20 12:55:58
```

Latency of data_read latency



Close

Job Queues

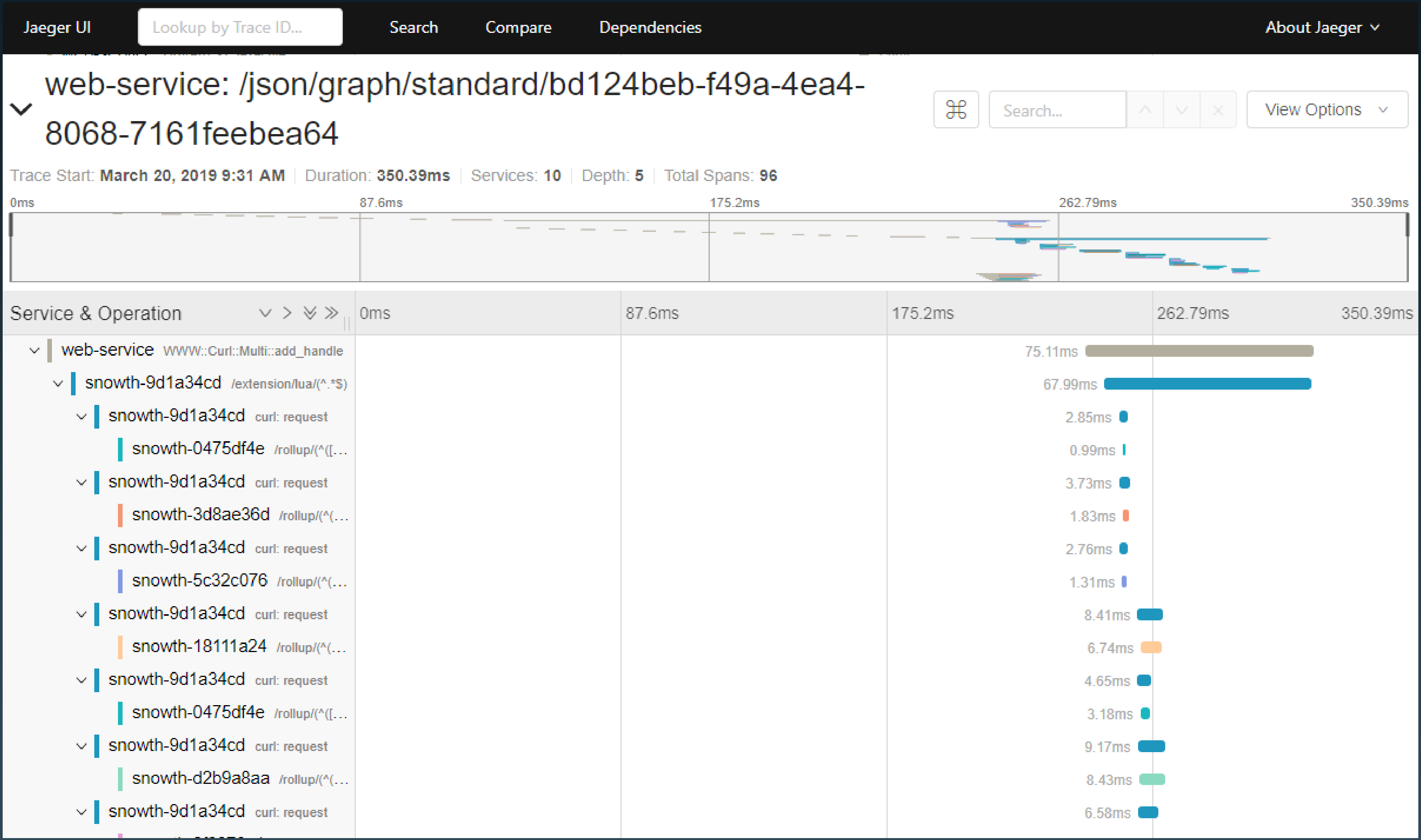
Used

	Queue					Running	
0	data_key_read					0.221ms	0
0	data_read					0.591ms	0
0	data_write					1.320ms	0
0	default_queue					17.678ms	0
0	journal-2f539d01-b137-4736-8797-911c4e1daube	4	9350543	0.026ms	0	0.137ms	0
0	journal-3c42054f-40e7-4f63-87e0-bb98329f6998	4	9350543	0.018ms	0	0.173ms	0
0	journal-4bb9202b-3a48-4408-8ac1-eb9723fccc17	4	9350543	0.030ms	0	0.177ms	0
0	journal-66a80c34-dfe5-4825-9cc4-eff981ecdaa1	4	9350543	0.011ms	0	0.121ms	0
0	journal-a212a487-25a5-47bd-8751-6330a4011677	4	9350543	0.044ms	0	0.200ms	0
0	journal-a53550b0-ebfd-431b-8662-92235ff2e733	4	9350543	0.021ms	0	0.127ms	0
0	journal-a5ee7c74-1147-4afa-a665-1951ae497aba	4	9350543	0.018ms	0	0.143ms	0
0	journal-be948fcc-c3ac-44f8-b5d3-e8292676bc40	4	9350543	0.028ms	0	0.140ms	0
0	journal-c483c214-9cd9-4cec-9f0c-8f4b46dfdb96	4	9350543	0.020ms	0	0.171ms	0

Known unknowns

Events & Distributed Tracing

A clearer story of what just happened.



Jaeger UI | Lookup by Trace ID... | Search | Compare | Dependencies | About Jaeger

web-service: /json/graph/standard/bd124beb-f49a-4ea4-8068-7161feebea64

Trace Start: **March 20, 2019 9:31 AM** | Duration: **350.39ms** | Services: **10** | Depth: **5** | Total Spans: **96**

Service & Operation	Duration
web-service WWW::Curl::Multi::add_handle	75.11ms
snowth-9d1a34cd /extension/lua/(^.*$)	67.99ms
snowth-9d1a34cd curl: request	2.85ms
snowth-0475df4e /rollup/(^([…	0.99ms
snowth-9d1a34cd curl: request	3.73ms
snowth-3d8ae36d /rollup/(^(…	1.83ms
snowth-9d1a34cd curl: request	2.76ms
snowth-5c32c076 /rollup/(^(…	1.31ms
snowth-9d1a34cd curl: request	8.41ms
snowth-18111a24 /rollup/(^(…	6.74ms
snowth-9d1a34cd curl: request	4.65ms
snowth-0475df4e /rollup/(^([…	3.18ms
snowth-9d1a34cd curl: request	9.17ms
snowth-d2b9a8aa /rollup/(^(…	8.43ms
snowth-9d1a34cd curl: request	6.58ms

Some stats...

- We only retain traces for a short period of time (up to about 3 days)
- We don’t trace with all detail on due to overhead
    - Full debugging on in a trace can produce up to 4Gb of trace data for a single user request
    - We do this sometimes via manual triggering as a debugging action
- Typically, between 10 and 2000 traces per request
- We use this as a debugging observability tool

During failure reconstruction, logs hold truth

Logging for humans

Computers talking to computers have better ways than logs. Logs are for computers talking to humans.

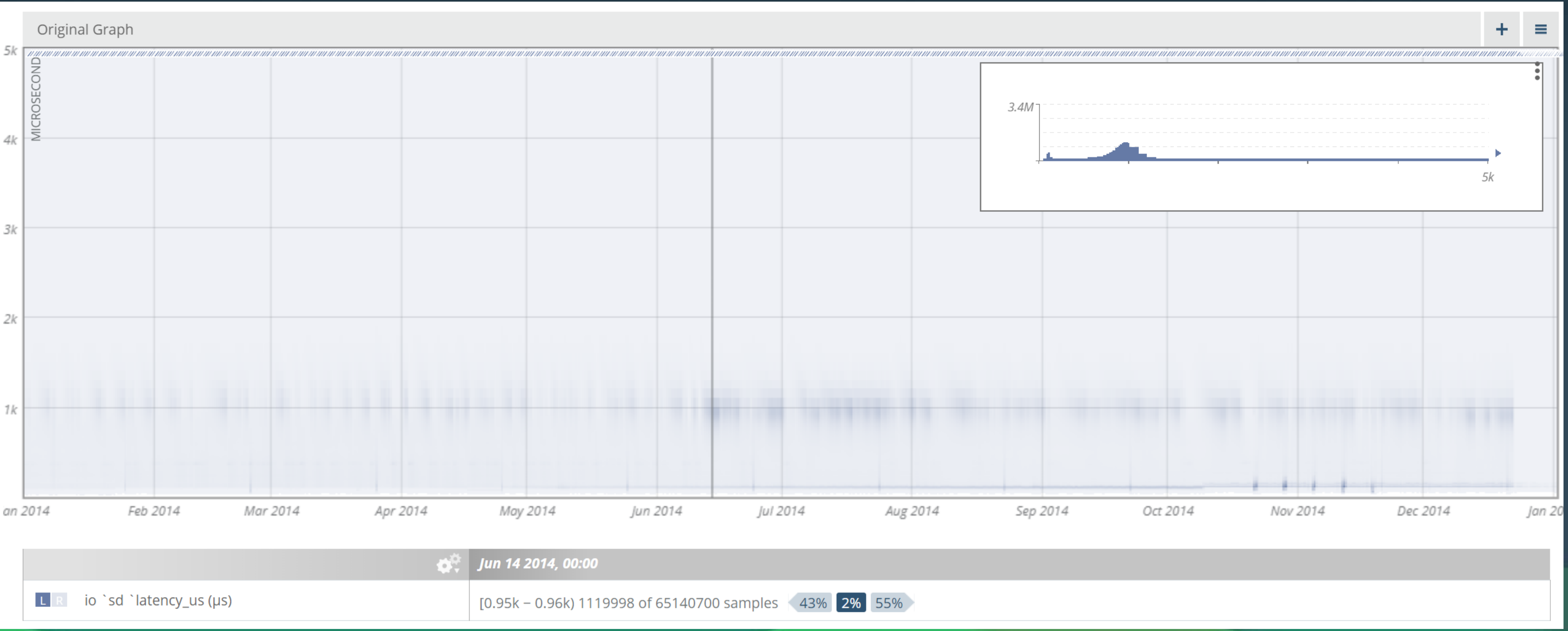
CIRCONUS

Real unknown unknowns are solved by:

Dynamic Tracing

eBPF / bpftrace
DTrace

IO Latency… single node… 2014



Jun 14 2014, 00:00

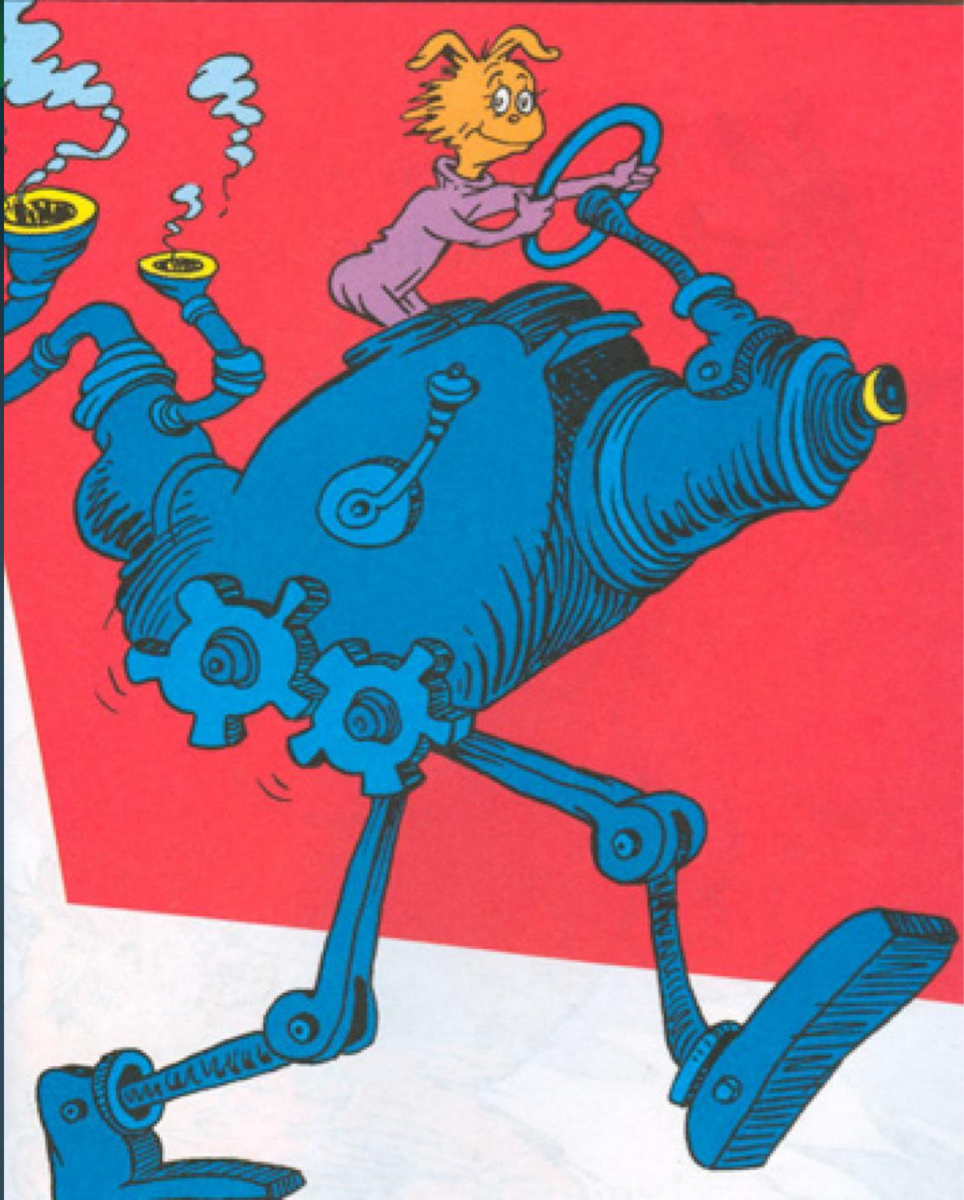
L R io `sd `latency_us (μs)

[0.95k – 0.96k) 1119998 of 65140700 samples 43% 2% 55%

Your m8g, o11y need to be accessible

Internalized MVP

No additional apparatus. No additional deployment constraints



CIRCONUS

Shipping software means more operators

Codify Operational Assessment & Procedures

More operators, less average knowledge. Ensure procedures are repeatable.
Tools -> Solutions

Every effort to bring SRE techniques to software engineering makes SRE more accessible and useful in Cloud/SaaS engineering.

CIRCONUS

Thank You.

Theo Schlossnagle

CEO, Circonus

@postwait