



Let's look at some traces

<http://bit.do/jaeger-hotrod>



Service dependencies diagram

Jaeger UI

Lookup by Trace ID...

Search

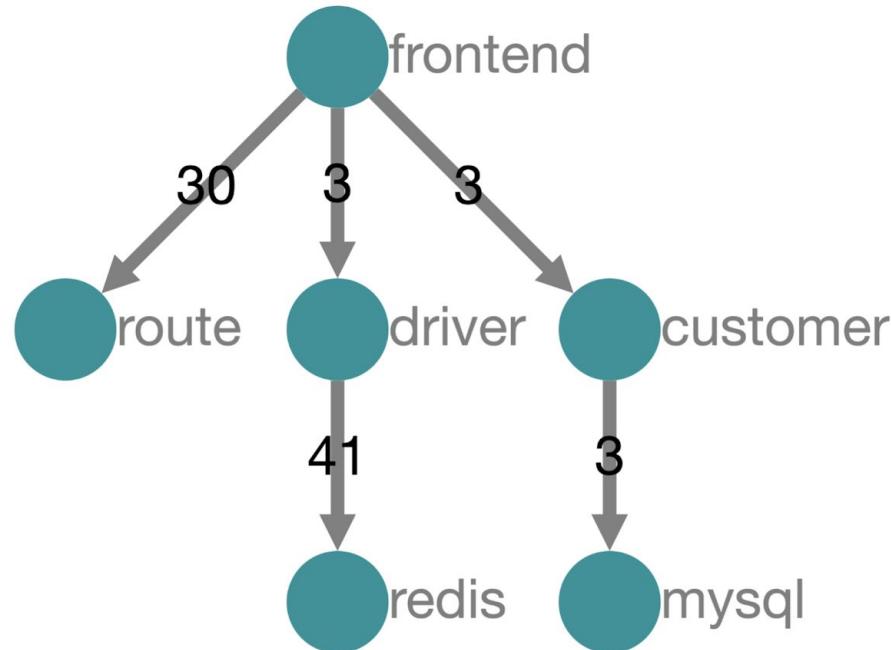
Compare

Dependencies

About Jaeger

Force Directed Graph

DAG



Transitive Service Graphs

4 Traces

Sort: Most Recent

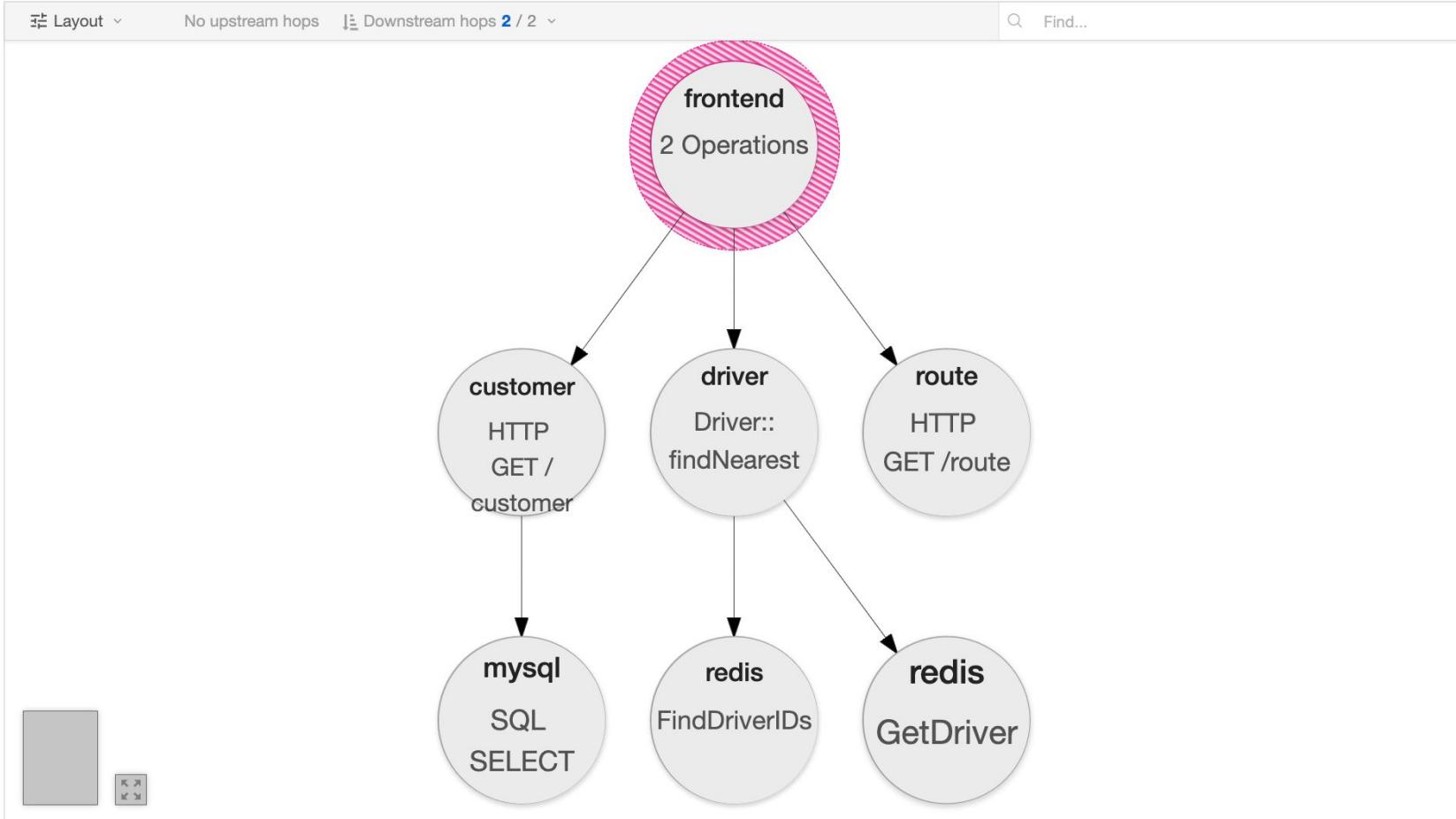
Deep Dependency Graph

Compare traces by selecting result items

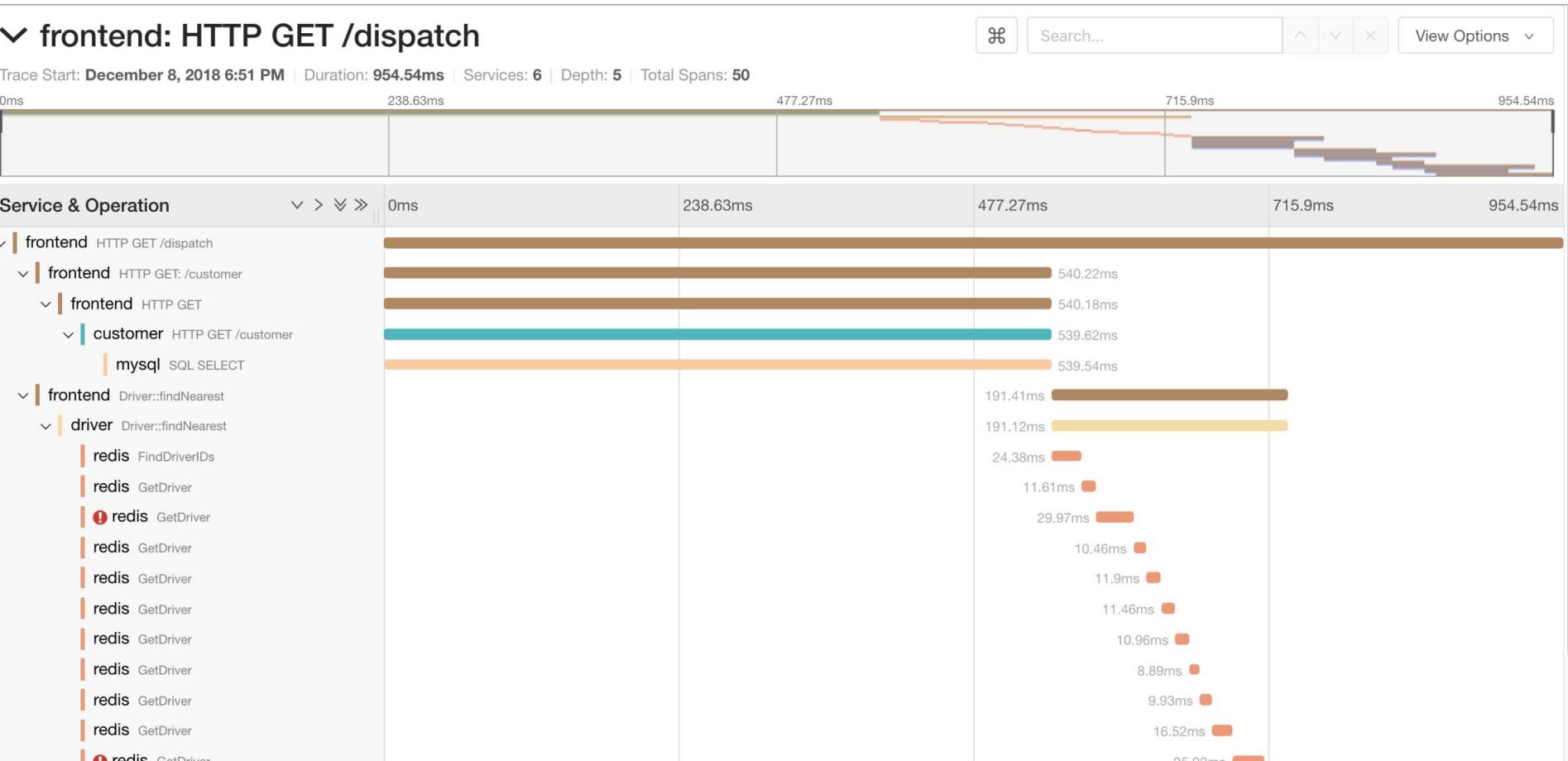


<input type="checkbox"/> frontend: HTTP GET /dispatch 3688087	1.04s
51 Spans 3 Errors customer (1) driver (1) frontend (24) mysql (1) redis (14) route (10)	Today 5:39:56 pm 5 minutes ago
<input type="checkbox"/> frontend: HTTP GET /dispatch 73e6e77	853.78ms
50 Spans 2 Errors customer (1) driver (1) frontend (24) mysql (1) redis (13) route (10)	Today 5:39:56 pm 5 minutes ago
<input type="checkbox"/> frontend: HTTP GET /dispatch d84845f	702.29ms
51 Spans 3 Errors customer (1) driver (1) frontend (24) mysql (1) redis (14) route (10)	Today 5:39:56 pm 5 minutes ago

Transitive Service Graphs



Trace timeline



Trace timeline – Parent → Child → Grandchild

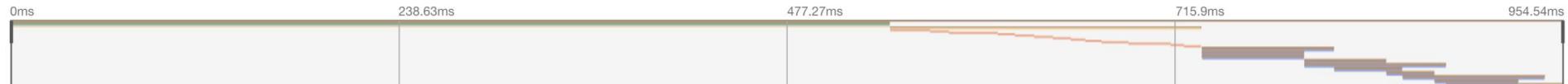
✓ frontend: HTTP GET /dispatch



Search...



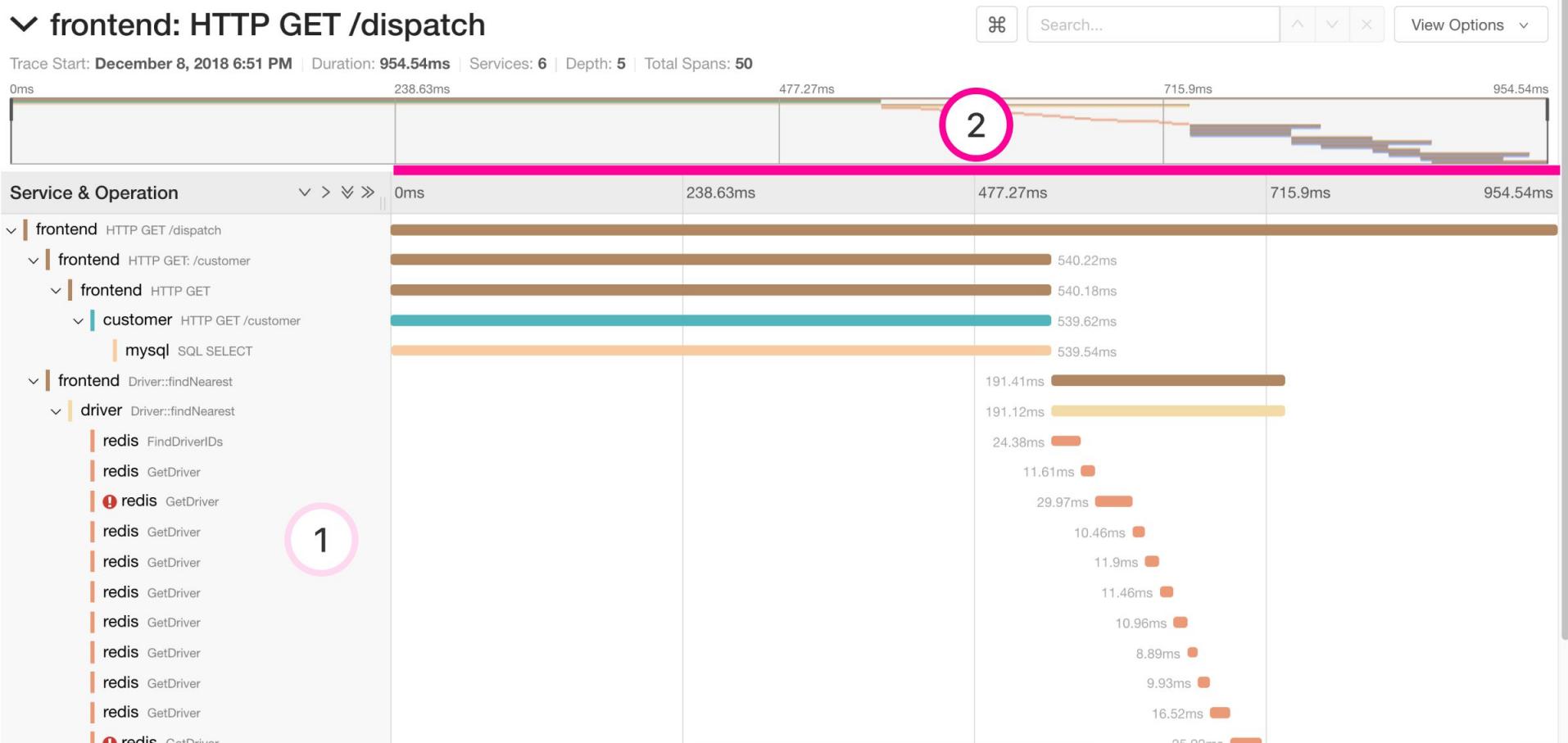
Trace Start: December 8, 2018 6:51 PM | Duration: 954.54ms | Services: 6 | Depth: 5 | Total Spans: 50



Service & Operation	0ms	238.63ms	477.27ms	715.9ms	954.54ms
✓ frontend HTTP GET /dispatch					
✓ frontend HTTP GET /customer				540.22ms	
✓ frontend HTTP GET				540.18ms	
✓ customer HTTP GET /customer			539.62ms		
mysql SQL SELECT			539.54ms		
✓ frontend Driver::findNearest				191.41ms	
✓ driver Driver::findNearest				191.12ms	
redis FindDriverIDs				24.38ms	
redis GetDriver				11.61ms	
redis GetDriver				29.97ms	
redis GetDriver				10.46ms	
redis GetDriver				11.9ms	
redis GetDriver				11.46ms	
redis GetDriver				10.96ms	
redis GetDriver				8.89ms	
redis GetDriver				9.93ms	
redis GetDriver				16.52ms	
redis GetDriver				25.92ms	

1

Trace timeline – Time + Mini-map



Trace timeline – A blocking operation

✓ frontend: HTTP GET /dispatch

⌘ Search... ▲ ▼ × View Options ▾

Trace Start: December 8, 2018 6:51 PM | Duration: 954.54ms | Services: 6 | Depth: 5 | Total Spans: 50



Service & Operation	0ms	238.63ms	477.27ms	715.9ms	954.54ms
✓ frontend HTTP GET /dispatch					
└ frontend HTTP GET :/customer				540.22ms	
└ frontend HTTP GET				540.18ms	
└ customer HTTP GET /customer			539.62ms		
└ mysql SQL SELECT			539.54ms		
└ frontend Driver::findNearest				191.41ms	
└ driver Driver::findNearest				191.12ms	
└ redis FindDriverIDs				24.38ms	
└ redis GetDriver				11.61ms	
└ redis GetDriver				29.97ms	
└ redis GetDriver				10.46ms	
└ redis GetDriver				11.9ms	
└ redis GetDriver				11.46ms	
└ redis GetDriver				10.96ms	
└ redis GetDriver				8.89ms	
└ redis GetDriver				9.93ms	
└ redis GetDriver				16.52ms	
└ redis GetDriver				25.92ms	

1

3

2

Trace timeline – Sequential operations

✓ frontend: HTTP GET /dispatch

⌘ Search... ▲ ▼ × View Options ▾

Trace Start: December 8, 2018 6:51 PM | Duration: 954.54ms | Services: 6 | Depth: 5 | Total Spans: 50



Service & Operation	0ms	238.63ms	477.27ms	715.9ms	954.54ms
✓ frontend HTTP GET /dispatch					
✓ frontend HTTP GET :/customer				540.22ms	
✓ frontend HTTP GET				540.18ms	
✓ customer HTTP GET /customer			539.62ms		
mysql SQL SELECT			539.54ms		
✓ frontend Driver::findNearest				191.41ms	
✓ driver Driver::findNearest				191.12ms	
redis FindDriverIDs				24.38ms	
redis GetDriver				11.61ms	
redis GetDriver				29.97ms	
redis GetDriver				10.46ms	
redis GetDriver				11.9ms	
redis GetDriver				11.46ms	
redis GetDriver				10.96ms	
redis GetDriver				8.89ms	
redis GetDriver				9.93ms	
redis GetDriver				16.52ms	
redis GetDriver				25.92ms	
redis GetDriver					

1

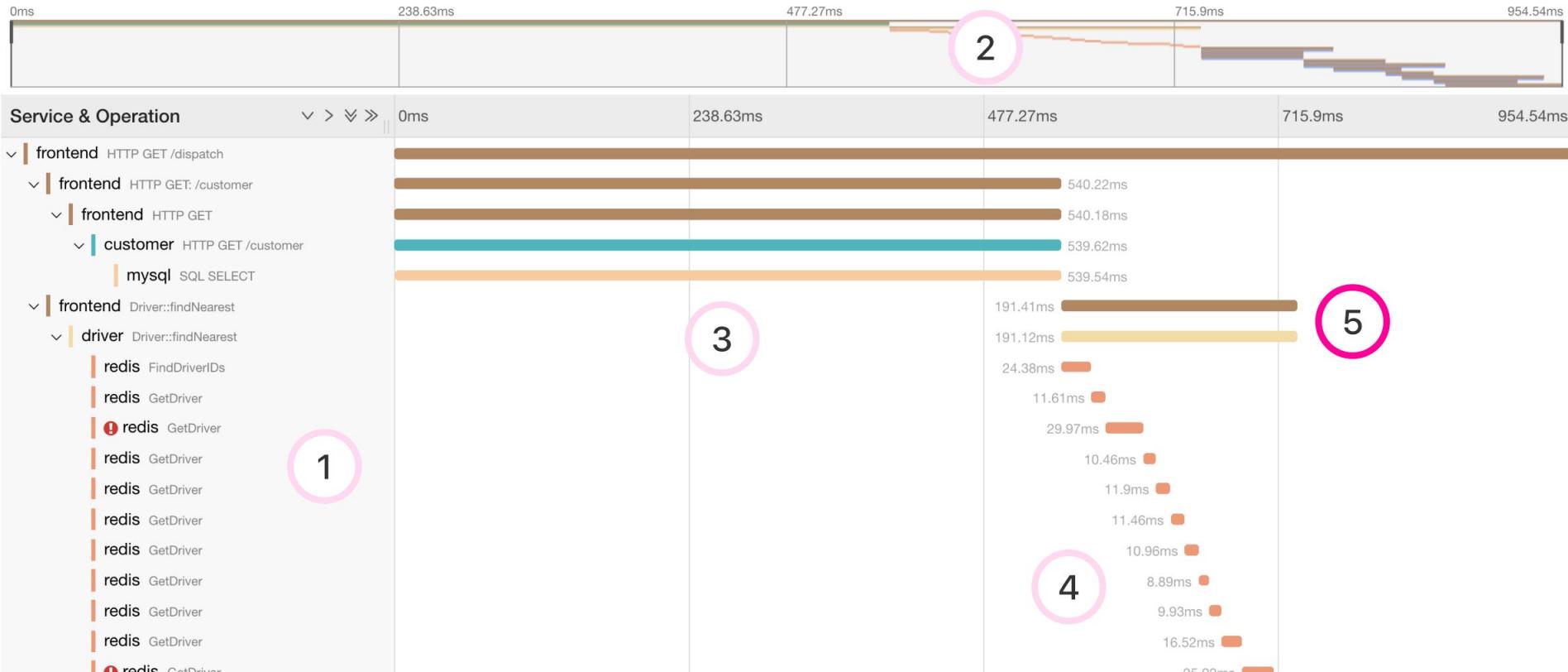
3

4

Trace timeline – Parents encompass descendants (generally)

✓ frontend: HTTP GET /dispatch

Trace Start: December 8, 2018 6:51 PM | Duration: 954.54ms | Services: 6 | Depth: 5 | Total Spans: 50



Span details

> frontend: HTTP GET /dispatch

Service & Operation 0ms 238.63ms 477.27ms 715.9ms 954.54ms

frontend HTTP GET /dispatch
frontend HTTP GET: /customer
 frontend HTTP GET
 customer HTTP GET /customer
mysql SQL SELECT

SQL SELECT Service: mysql | Duration: 539.54ms | Start Time: 0.67ms

Tags

span.kind	"client"
peer.service	"mysql"
sql.query	"SELECT * FROM customer WHERE customer_id=392"
request	"3878-3"

Process: client-uuid = 55627059ae2defbd | hostname = joef-C02TX0LYHTDG | ip = 192.168.1.5 | jaeger.version = Go-2.15.0

Logs (2)

0.68ms: event = Waiting for lock behind 2 transactions blockers = [3878-1 3878-2]
282.29ms: event = Acquired lock with 0 transactions waiting behind

Log timestamps are relative to the start time of the full trace.

SpanID: 7aecd811f9df684

frontend Driver::findNearest
driver Driver::findNearest

Span details – Database query

> frontend: HTTP GET /dispatch

Service & Operation 0ms 238.63ms 477.27ms 715.9ms 954.54ms

frontend HTTP GET /dispatch
frontend HTTP GET: /customer
 frontend HTTP GET
 customer HTTP GET /customer
mysql SQL SELECT

SQL SELECT Service: mysql | Duration: 539.54ms | Start Time: 0.67ms

Tags

span.kind	"client"
peer.service	"mysql"
sql.query	"SELECT * FROM customer WHERE customer_id=392"
request	"3878-3"

1

Process: client-uuid = 55627059ae2defbd | hostname = joef-C02TX0LYHTDG | ip = 192.168.1.5 | jaeger.version = Go-2.15.0

Logs (2)

0.68ms: event = Waiting for lock behind 2 transactions blockers = [3878-1 3878-2]
282.29ms: event = Acquired lock with 0 transactions waiting behind

Log timestamps are relative to the start time of the full trace.

SpanID: 7aecad811f9df684

frontend Driver::findNearest
driver Driver::findNearest

191.41ms
191.12ms

Span details – Lock contention

> frontend: HTTP GET /dispatch

Service & Operation 0ms 238.63ms 477.27ms 715.9ms 954.54ms

- frontend HTTP GET /dispatch
- frontend HTTP GET: /customer
 - frontend HTTP GET
 - customer HTTP GET /customer
- mysql SQL SELECT

SQL SELECT Service: mysql | Duration: 539.54ms | Start Time: 0.67ms

Tags

span.kind	"client"
peer.service	"mysql"
sql.query	"SELECT * FROM customer WHERE customer_id=392"
request	"3878-3"

1

Process: client-uuid = 55627059ae2defbd | hostname = joef-C02TX0LYHTDG | ip = 192.168.1.5 | jaeger.version = Go-2.15.0

Logs (2)

0.68ms: event = Waiting for lock behind 2 transactions blockers = [3878-1 3878-2]

282.29ms: event = Acquired lock with 0 transactions waiting behind

Log timestamps are relative to the start time of the full trace.

2

SpanID: 7aecd811f9df684

- frontend Driver::findNearest
- driver Driver::findNearest

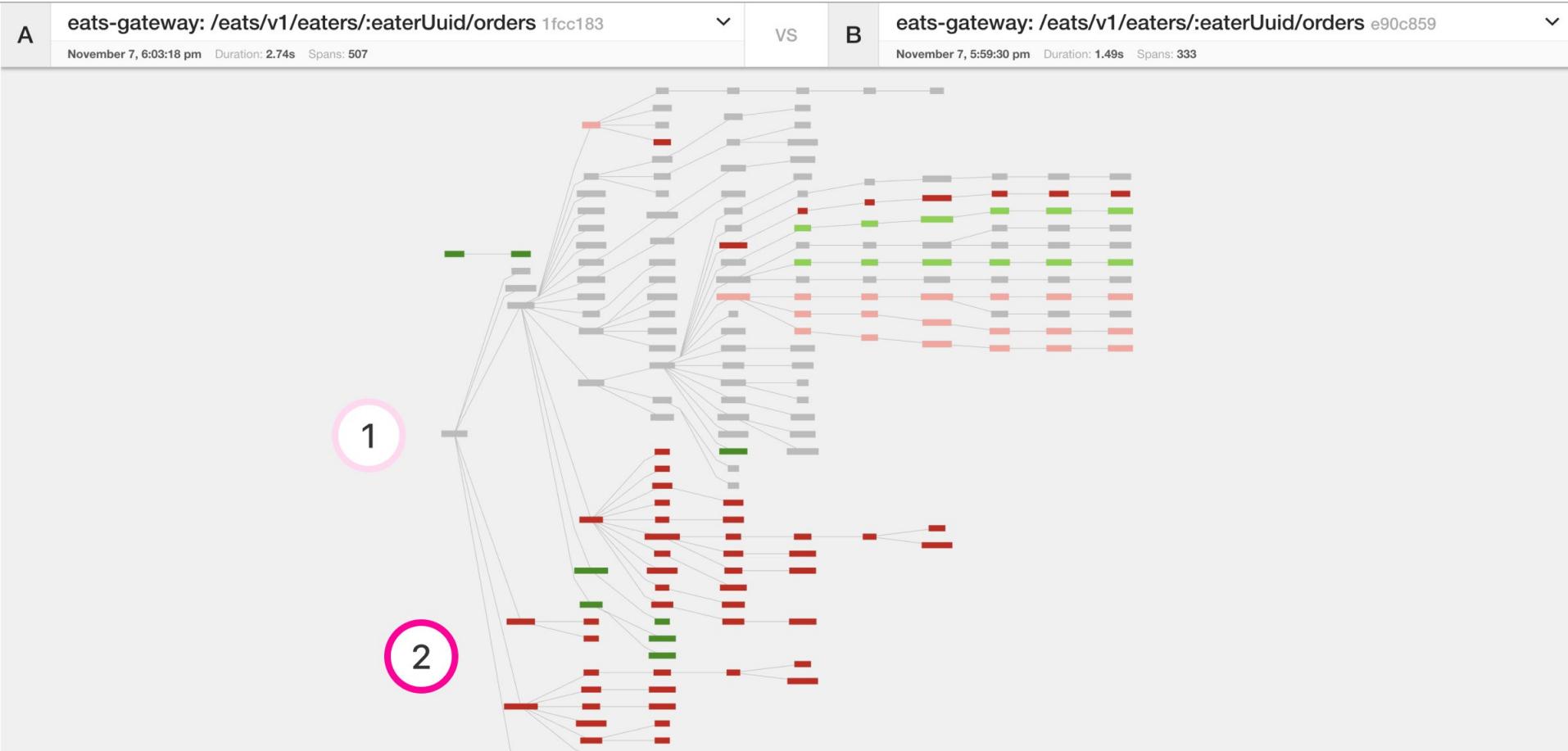
Comparing trace structures – Unified diff



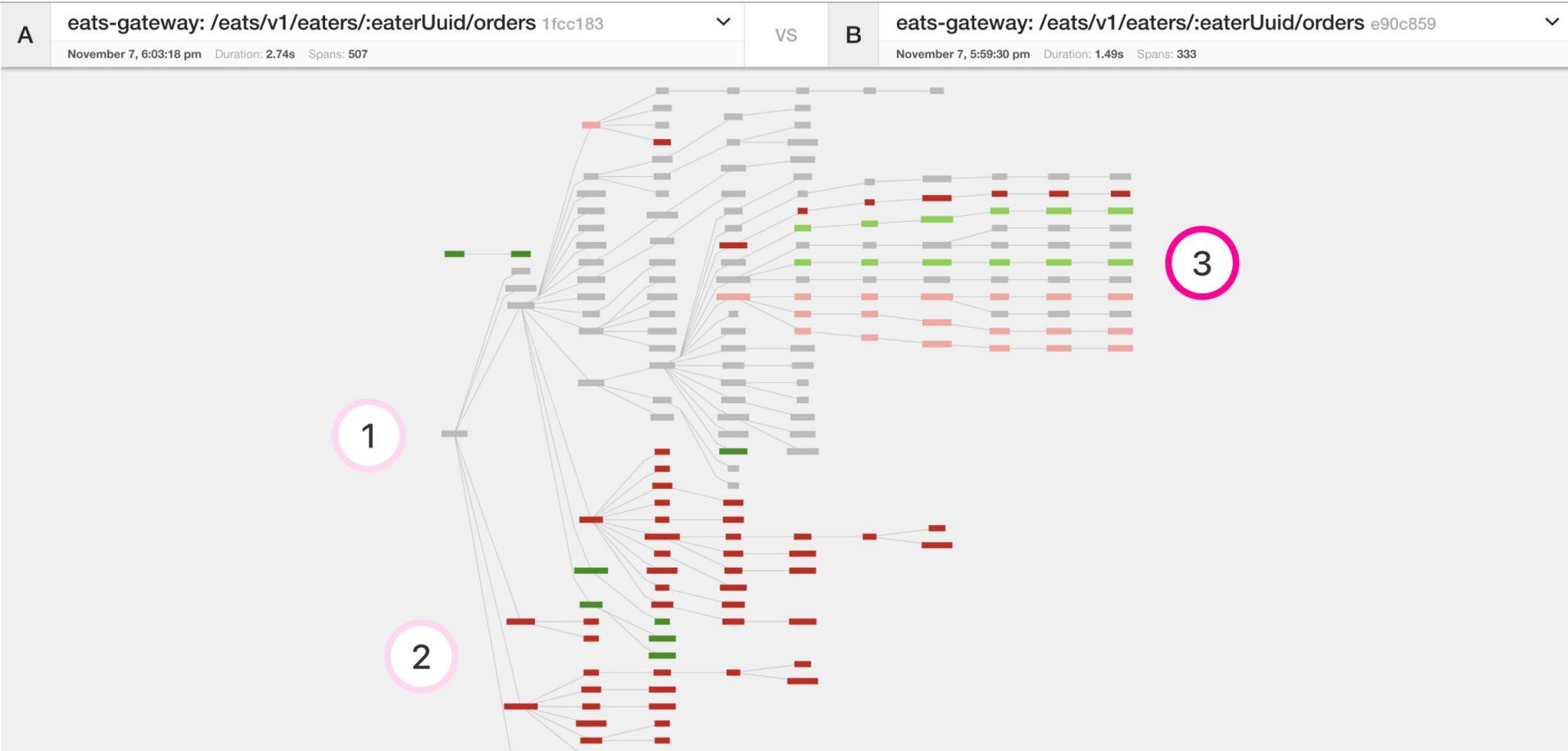
Comparing trace structures – Shared structure



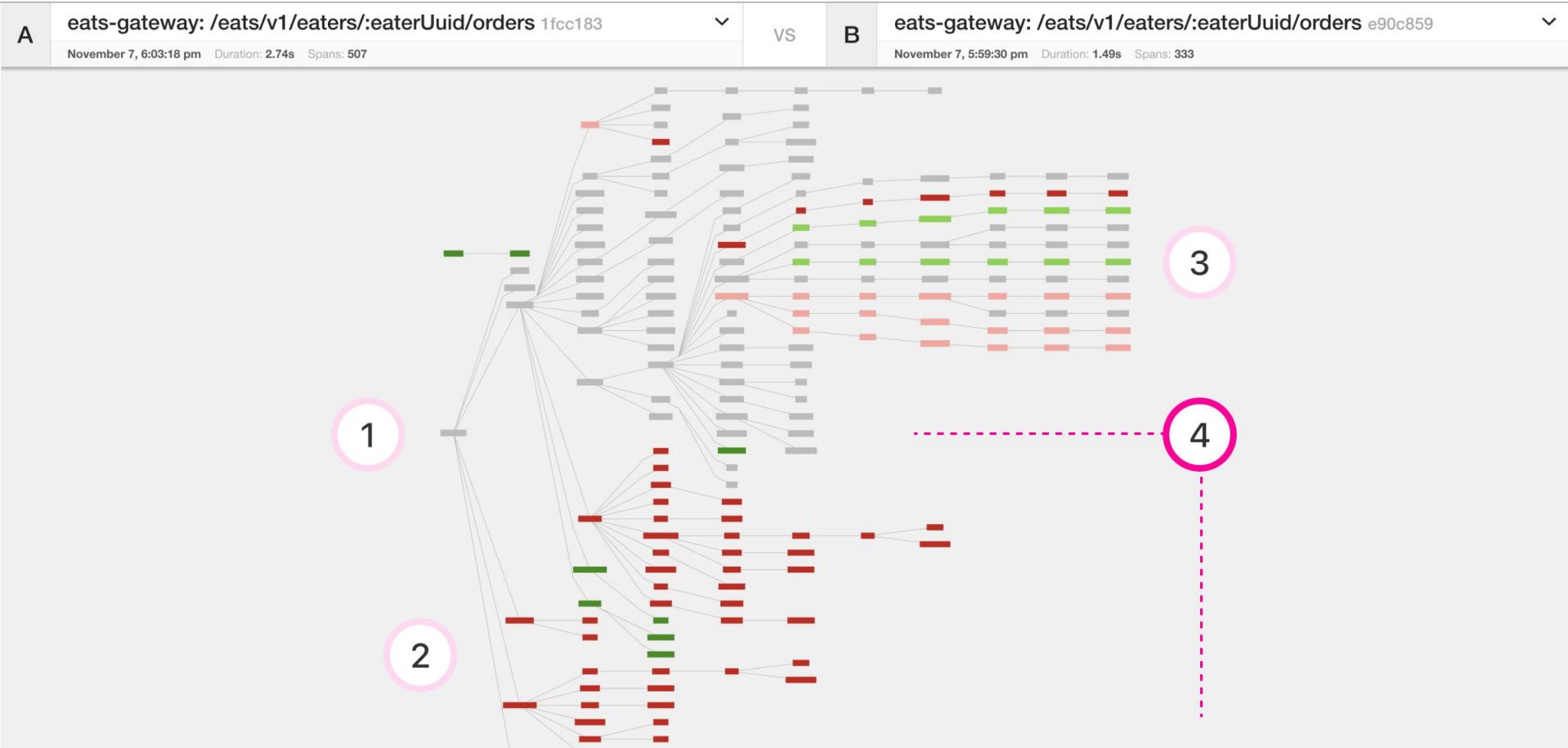
Comparing trace structures – Absent in one or the traces



Comparing trace structures – More or less within a node



Comparing trace structures – Substantial divergence



"You have an outstanding balance..."

> eats-gateway: /eats/v1/eaters/:eaterUuid/orders

Service & Operation 0ms 371.25ms 742.5ms 1.11s 1.49s

Service: eats-gateway | Duration: 1.29s | Start Time: 192ms

Logs (1)

1.48s

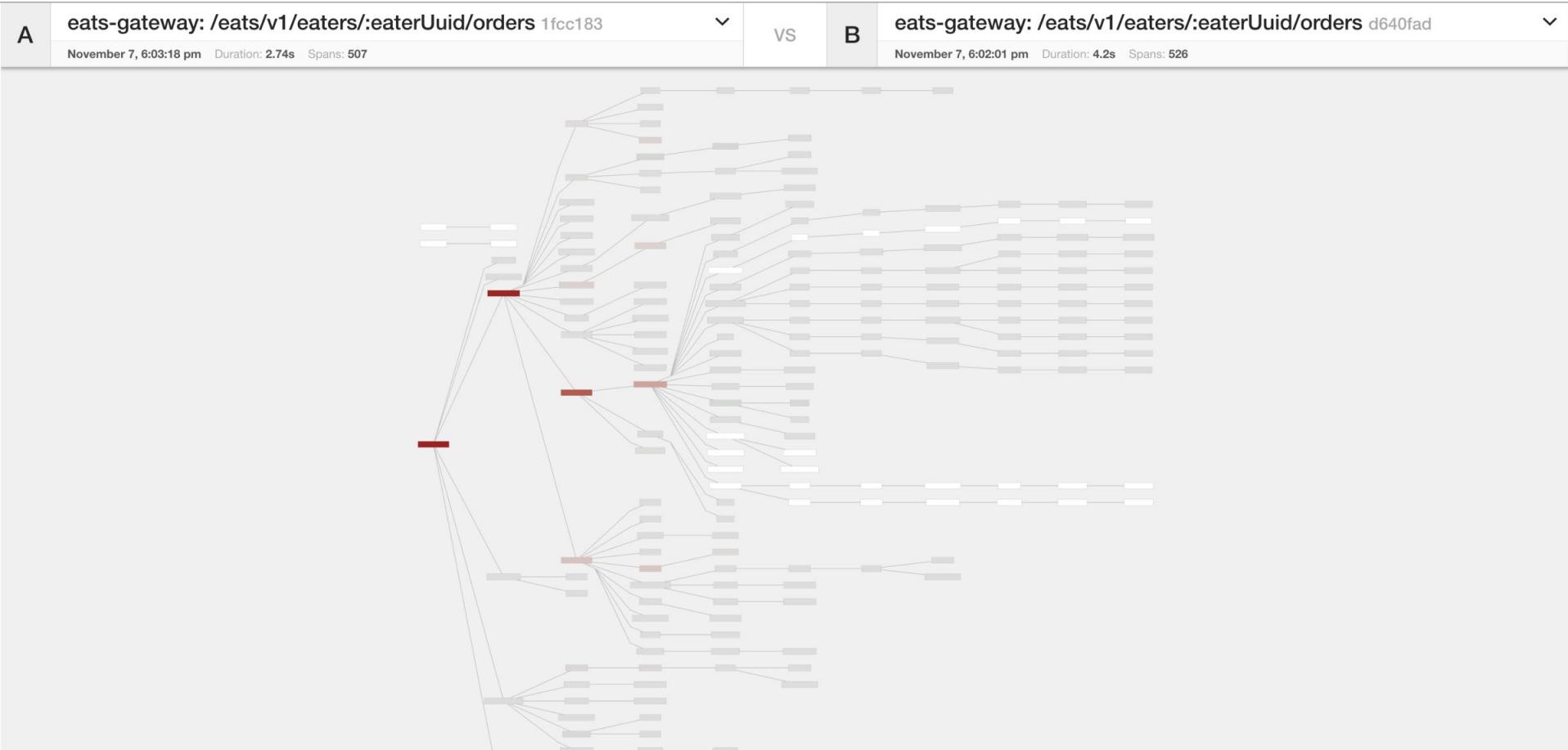
event "error"
error.kind "TChannelError"
error.object {
 info: {
 message: "Please verify payment information to secure your account",
 statusCode: 403,
 shouldRetry: false,
 stack: "***errors.errorString You have an outstanding balance due to a credit card problem. Please update your billing settings.
/there/are/many/pathes/up/the/mountain:150 (0x1337b0)
/there/are/many/pathes/up/the/mountain:74 (0x1337b0)
/there/are/many/pathes/up/the/mountain:83 (0x1337b0)
/there/are/many/pathes/up/the/mountain:118 (0x1337b0)
/there/are/many/pathes/up/the/mountain:71 (0x1337b0)
/there/are/many/pathes/up/the/mountain:36 (0x1337b0)
/there/are/many/pathes/up/the/mountain:22 (0x1337b0)
/there/are/many/pathes/up/the/mountain:729 (0x1337b0)
/there/are/many/pathes/up/the/mountain:470 (0x1337b0)
/there/are/many/pathes/up/the/mountain:458 (0x1337b0)
/there/are/many/pathes/up/the/mountain:1269 (0x1337b0)
/there/are/many/pathes/up/the/mountain:1030 (0x1337b0)
/there/are/many/pathes/up/the/mountain:94 (0x1337b0)
/there/are/many/pathes/up/the/mountain:163 (0x1337b0)
/there/are/many/pathes/up/the/mountain:237 (0x1337b0)
/there/are/many/pathes/up/the/mountain:118 (0x1337b0)"
 }
}

1

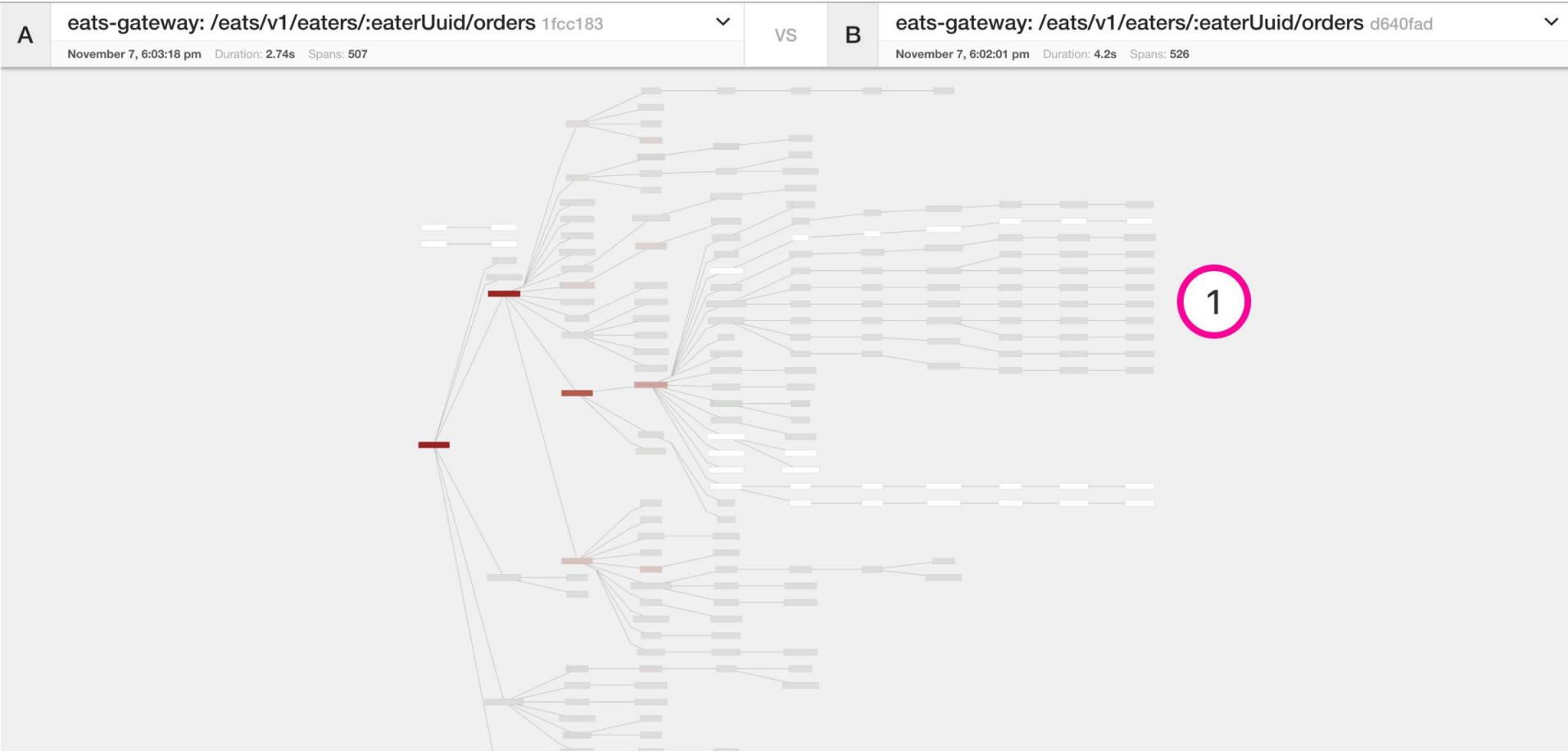
Log timestamps are relative to the start time of the full trace.

SpanID: 63bd06b7a7ed85b4

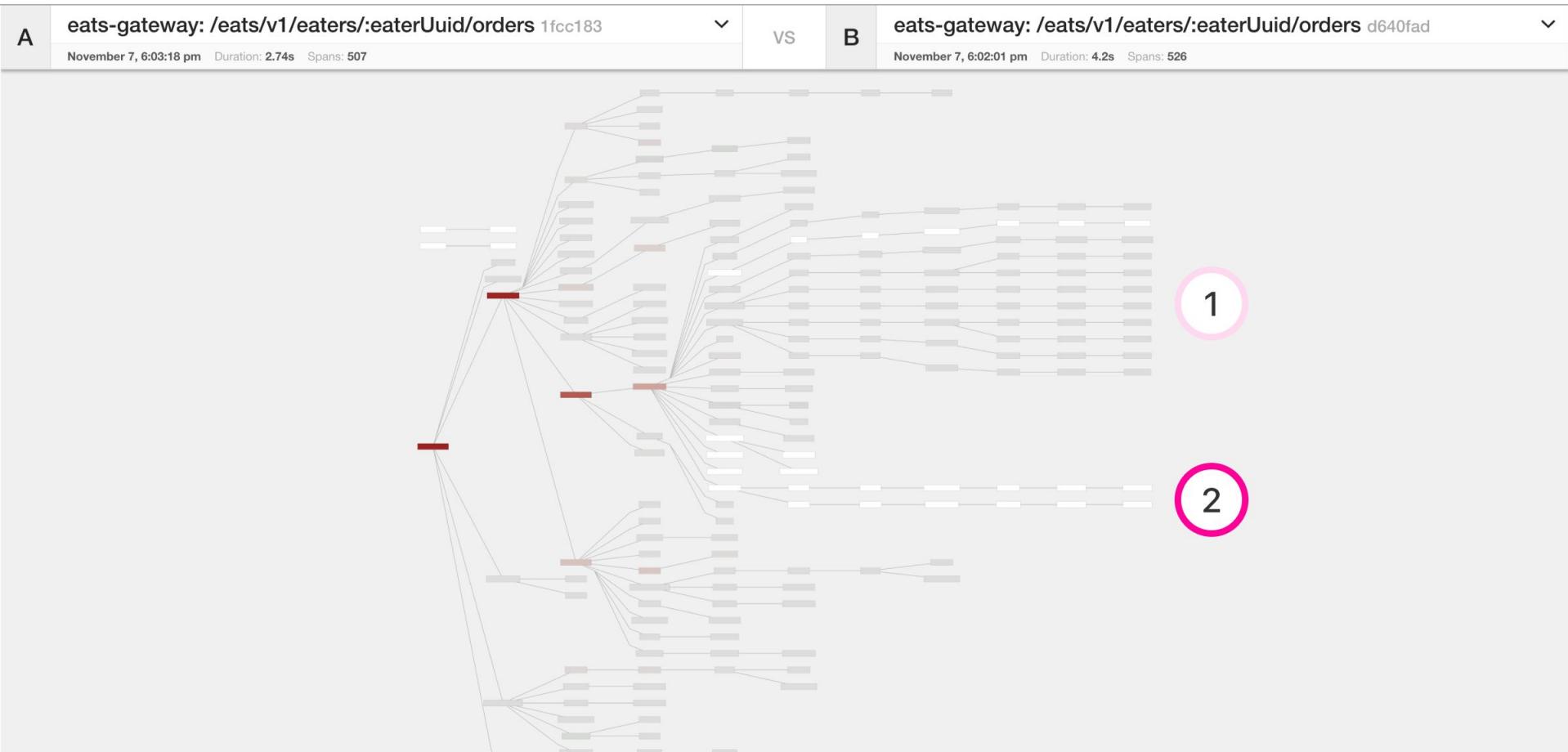
Comparing span durations



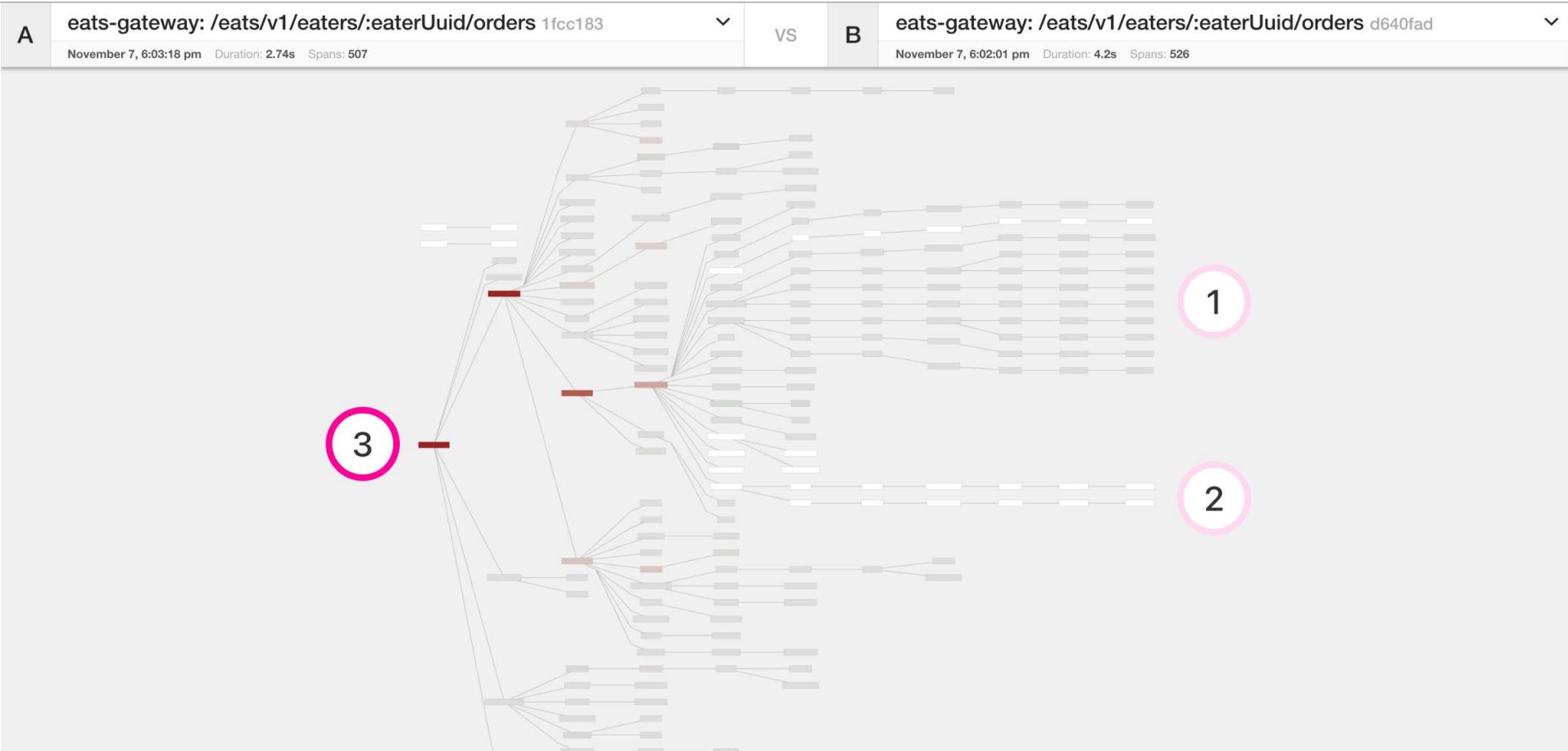
Comparing span durations – Similar durations



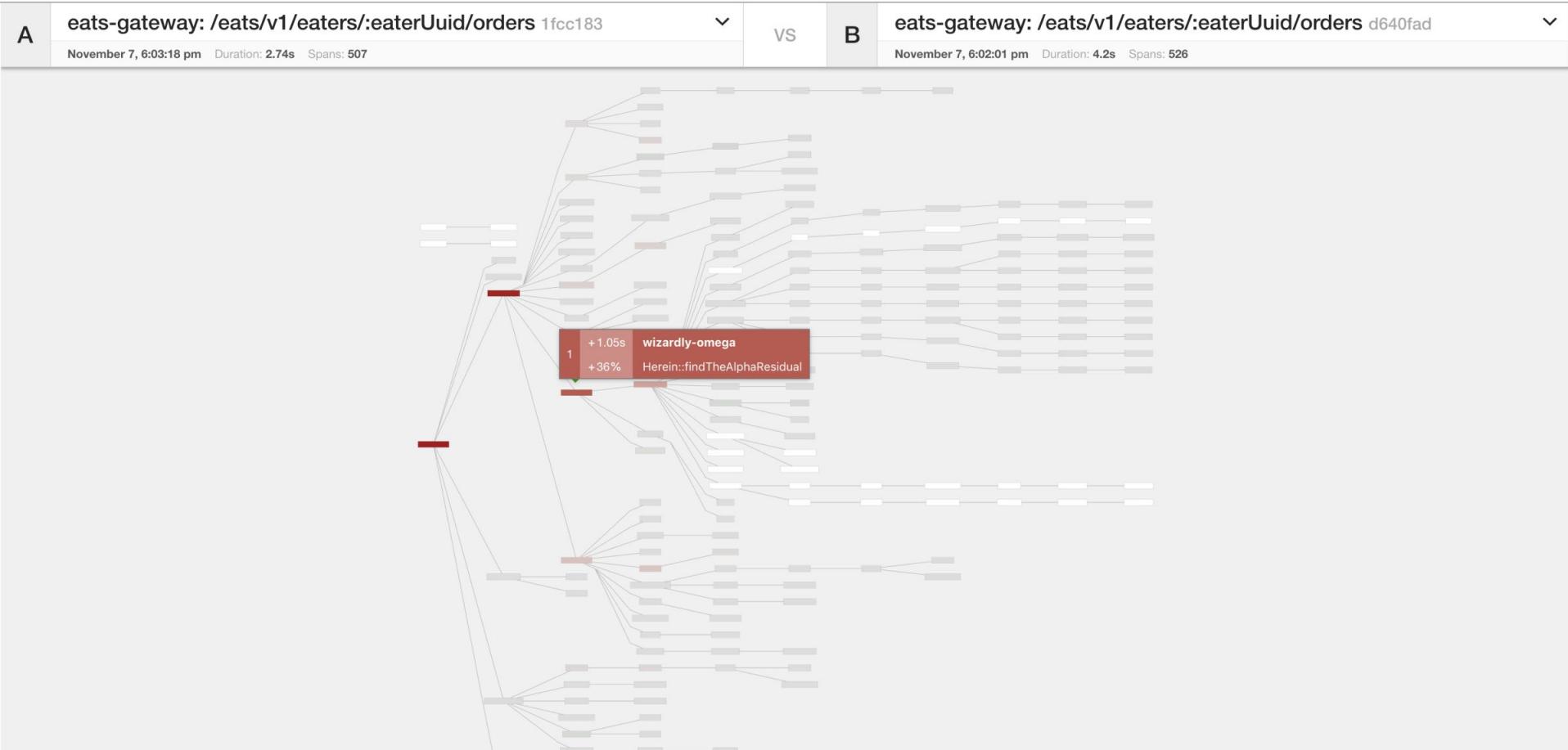
Comparing span durations – Nodes that aren't shared



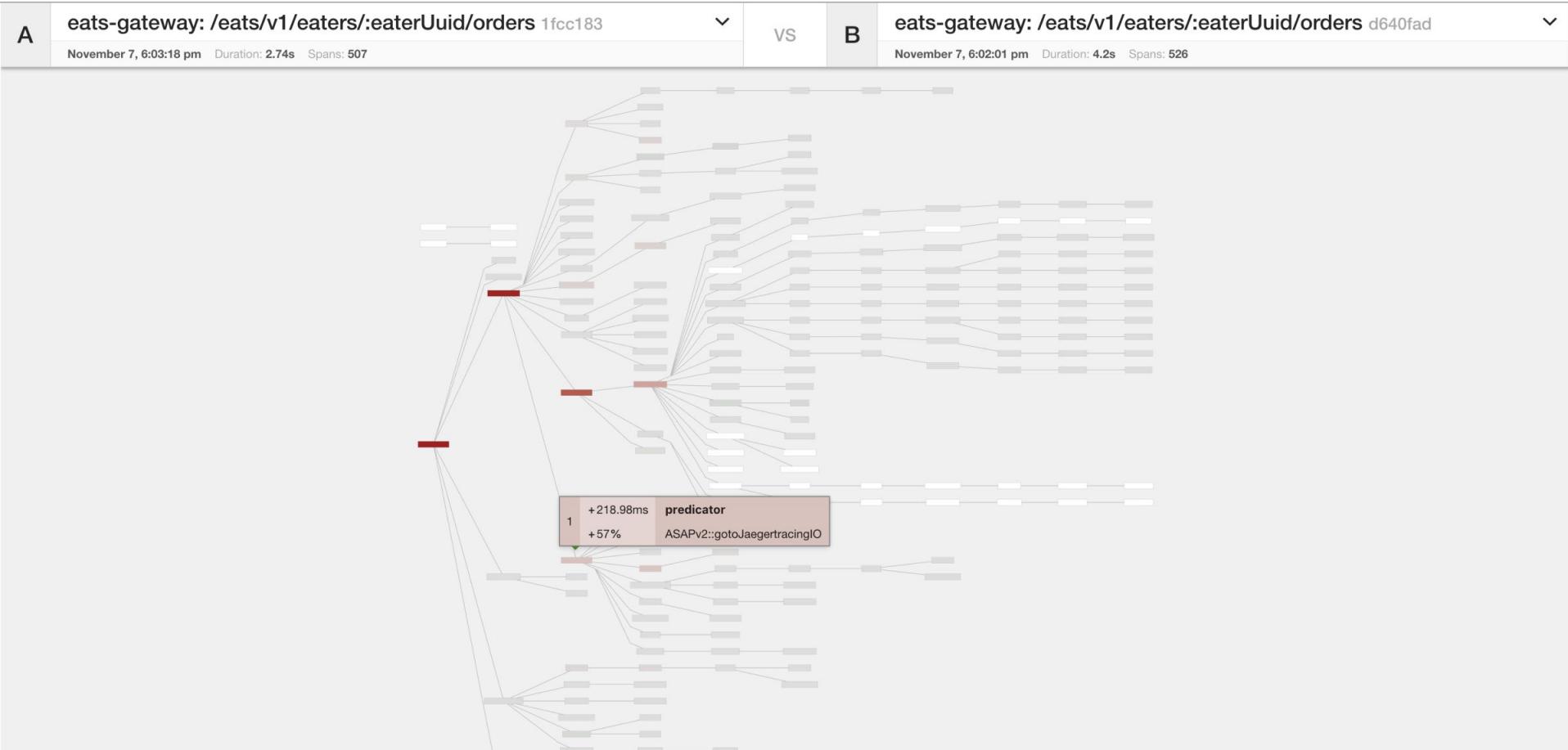
Comparing span durations – Follow the slower nodes



Comparing span durations



Comparing span durations



Graph Visualizations

Gantt chart is not great for traces with many 100s of spans

- Trace Diffs
 - Compare two traces
 - Compare one trace against a group of traces (coming soon)
- Trace Graph
 - Call graph visualization with mini-aggregations
 - Showing paths rather than individual RPCs

Graph Visualizations

- Surface less information
- Condense the structural representation
- Emphasize the differences
- Distinct comparison modes simplify the comparisons

Distributed Tracing Systems

distributed
transaction
monitoring

performance
and latency
optimization

root cause
analysis

service
dependency
analysis

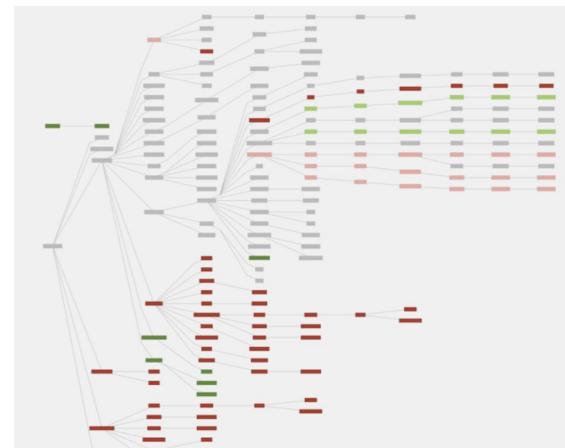
distributed context propagation



Jaeger

Architecture

Jaeger, a Distributed Tracing Platform



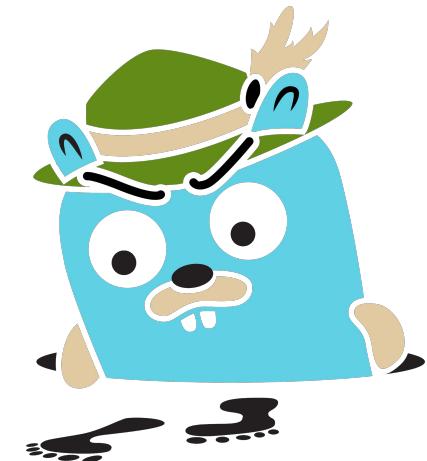
Instrumentation not included

Jaeger project does not provide instrumentation!

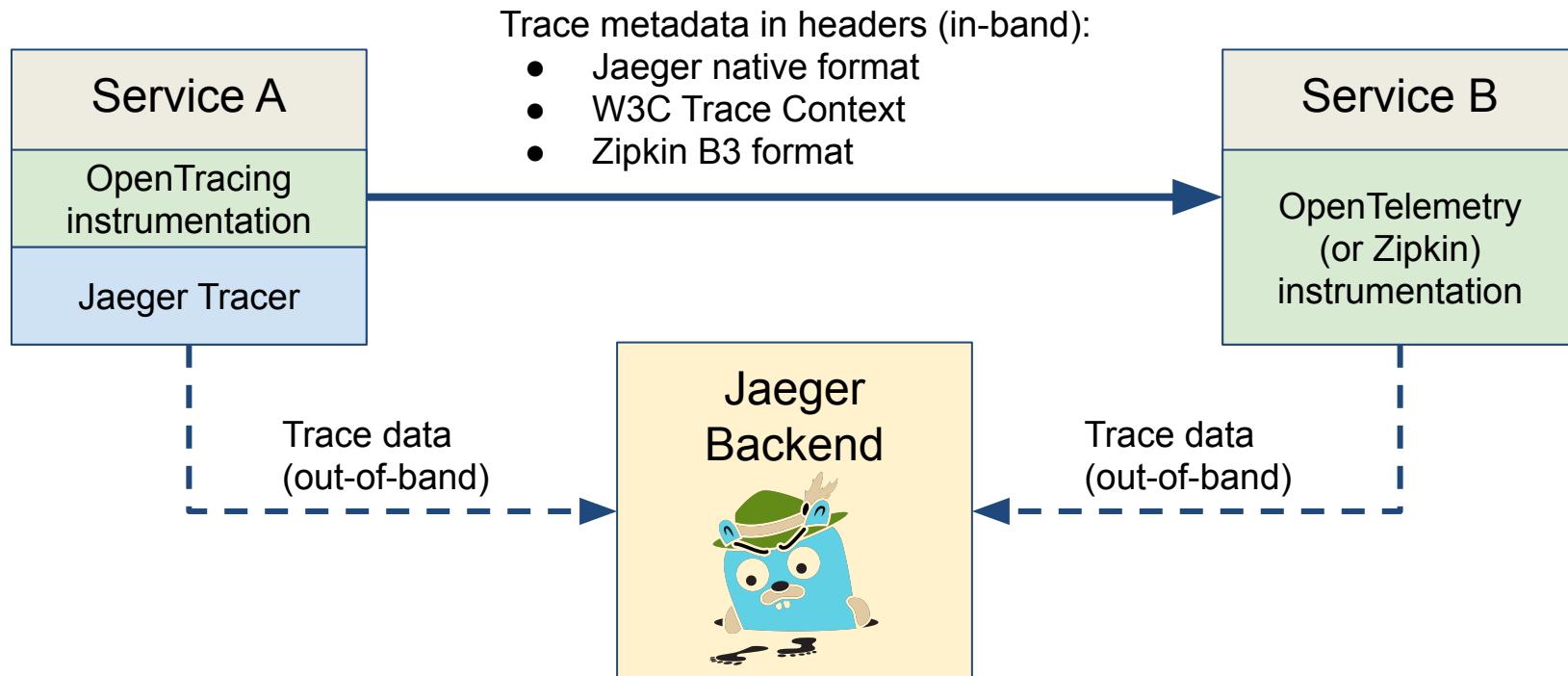
Use OpenTracing or OpenTelemetry.

Jaeger - /'yāgər/, noun: hunter

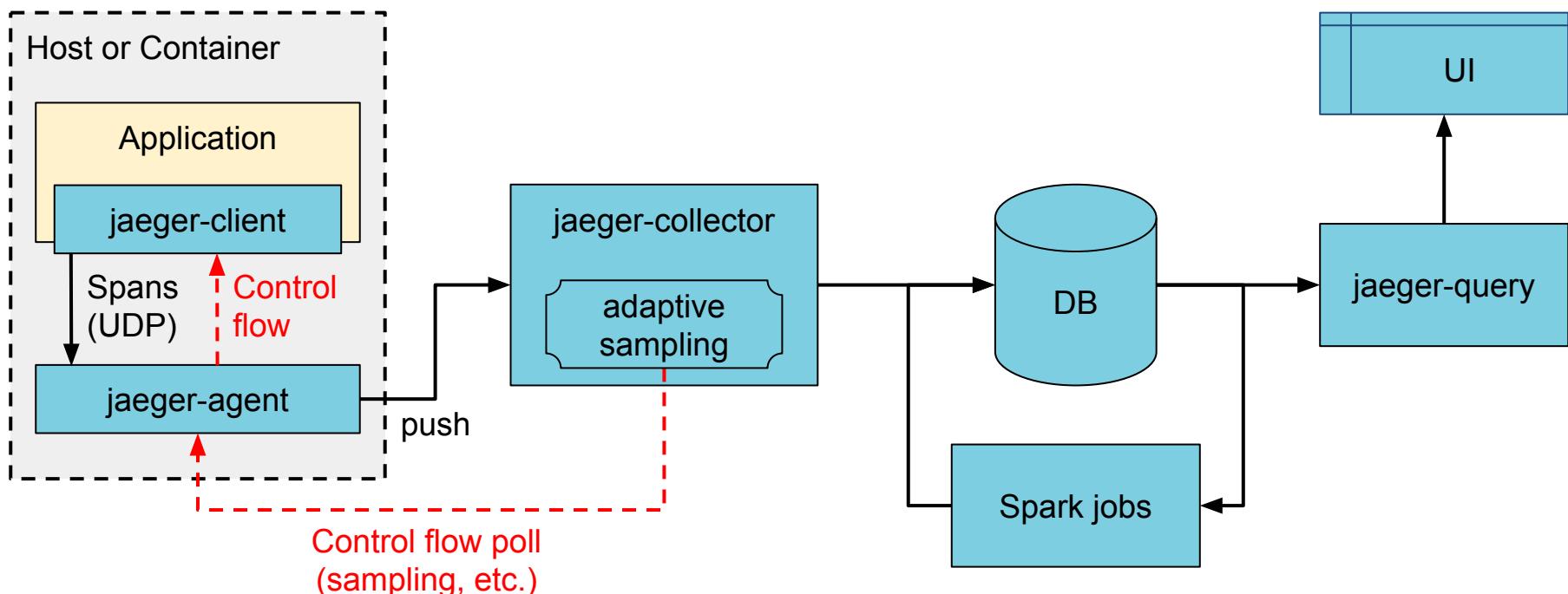
- Inspired by Google's Dapper and OpenZipkin
- Created at Uber in August 2015 ([blog](#))
- Open sourced in April 2017
- Joined CNCF in Sep 2017 (as incubating)
- Graduated to top-level CNCF project Oct 2019 ([CNCF announcement](#))



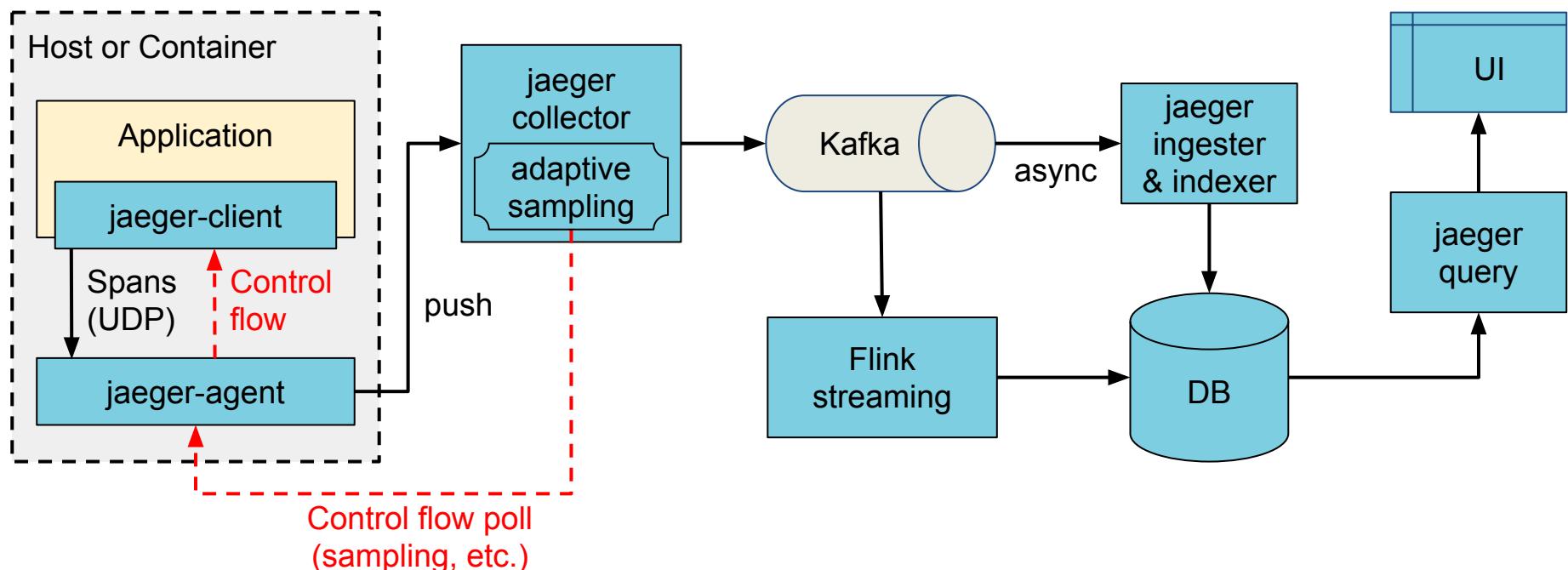
Jaeger and your system



Architecture 2017: Push



Architecture now: Push+Async+Streaming



Technology Stack

- Go backend
- Pluggable storage
 - Cassandra, Elasticsearch, badger, memory
- React/Javascript frontend
- OpenTracing Instrumentation libraries
- Integration with Kafka, Apache Flink



Go



Java™
POWERED



python
powered



Zipkin Compatibility

- Clients
 - Zipkin B3-*** headers for context propagation
 - Interop between Jaeger-instrumented and Zipkin-instrumented apps
- Collector
 - Zipkin Thrift, Protobuf, and JSON v2 span format
 - Use Zipkin instrumentation (e.g. Brave) to send traces to Jaeger
- Kafka



Jaeger

And Sampling



Why do we sample

- Tracing data is quite rich (2 spans per RPC)
- Saving everything incurs large storage costs
- Performance overhead from instrumentation

Head-based (upfront) sampling

- Sampling decision is made at the start of the trace propagated in the trace context
- Minimal perf overhead when trace is not sampled
- Easy to implement, supported by Jaeger SDKs
- Can easily miss rare anomalies/outliers

Head-based sampling in Jaeger

- SDKs can be configured with different samplers (probabilistic, rate limiting, etc.)
- SDKs default to “remote” sampler that polls the actual sampling strategy from the backend, which allows centralized configuration
- Configuration is per service & endpoint

Tail-based (post-trace) sampling

- Sampling decision is made at the end of the trace
- Can be much more intelligent, based on observed latency, errors, etc. Can catch anomalies.
- Requires temporary storage of all traces
- Incurs perf overhead even for traces that may be later discarded

Tail-based sampling in Jaeger

- Supported in `jaeger-opentelemetry-collector`
- Configurable sampling rules: latency, certain tags
- Single-node mode only, multi-node sharded solution will be available in the future

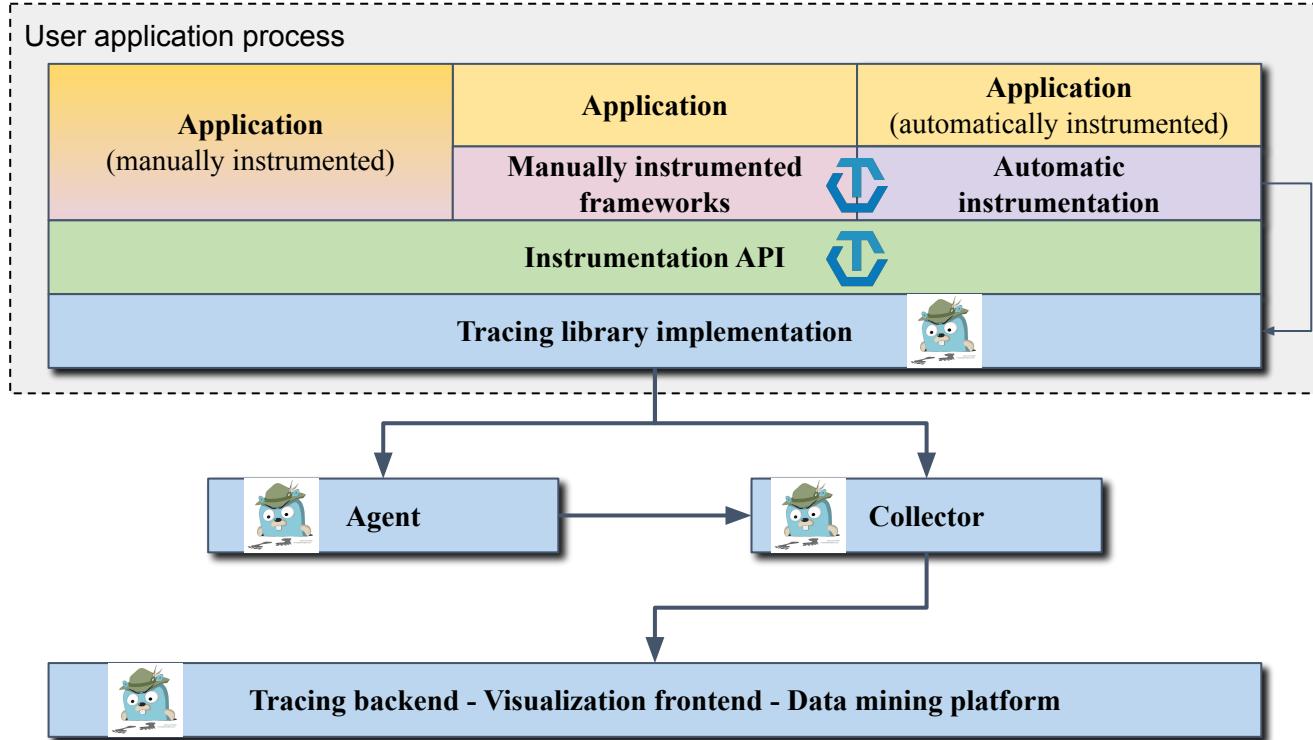


Jaeger

And OpenTelemetry



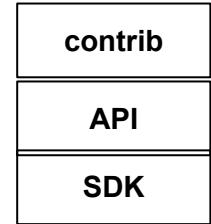
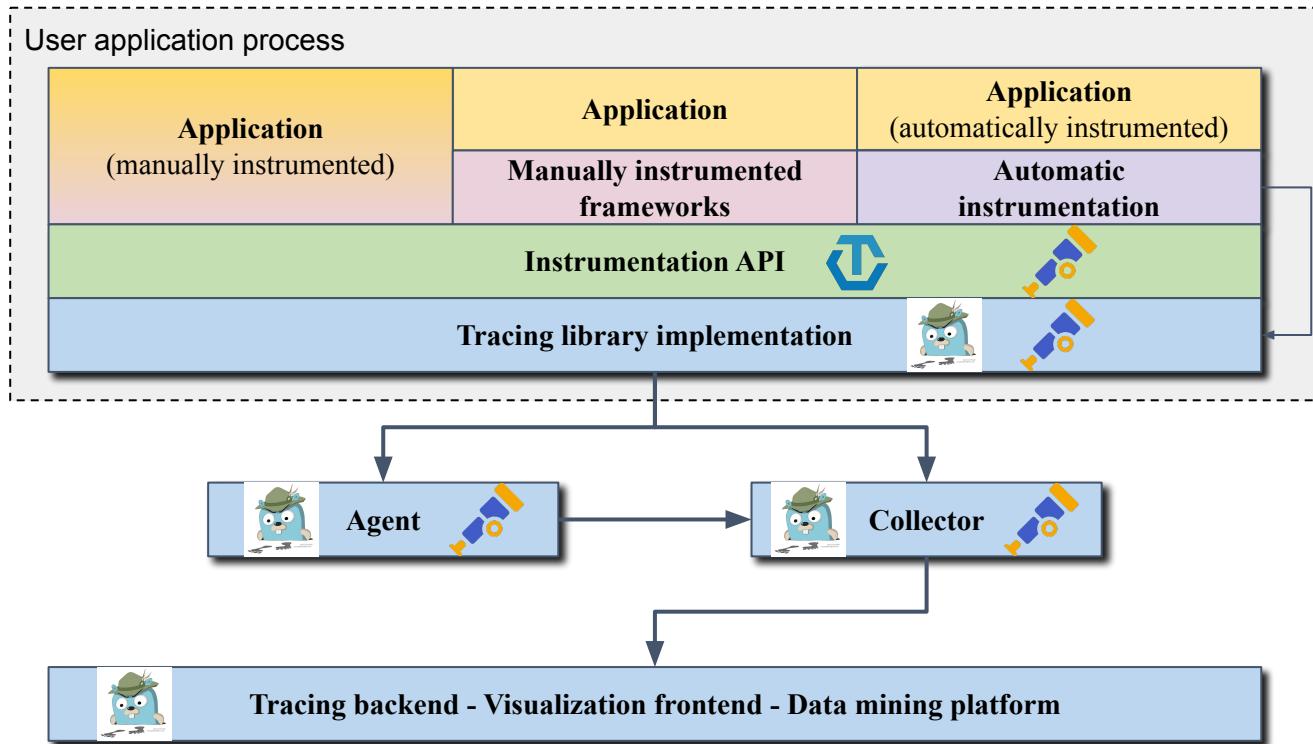
Jaeger with OpenTracing



Jaeger components on OpenTelemetry

- OpenTelemetry Collector is written in Go
- We built Jaeger-specific versions
 - Have the same capabilities as upstream OTEL
 - With Jaeger extensions, e.g. storage
- We're converting Jaeger storage implementation to OTEL data model for better compatibility

Jaeger with OpenTelemetry



New Features

- Kubernetes Operator
- Badger storage
- Storage plugins: Couchbase, InfluxDB
- Visual trace comparisons
- Security improvements
 - TLS with gRPC, Kafka, Elasticsearch

Documentation Website

- Releases & Downloads
- Architecture
- Deployment
- Command line options
- Client features

To learn more

- Jaeger Deep Dive - Wed, Aug 19, 13:45 - 14:20



<https://jaegertracing.io>

Getting in Touch

- GitHub: <https://github.com/jaegertracing>
- Chat: <https://gitter.im/jaegertracing/>
- Mailing List - jaeger-tracing@googlegroups.com
- Blog: <https://medium.com/jaegertracing>
- Twitter: <https://twitter.com/JaegerTracing>
- Bi-Weekly Community Meetings



Learn More

Website: jaegertracing.io/

Blog: medium.com/jaegertracing