CLOUD NATIVE
COMPUTING FOUNDATION

Jaeger
Project Deep Dive

Pavol Loffay (Red Hat), Joe Farro (Uber),
Yuri Shkuro (Uber)

CloudNativeCon NA, Seattle, Dec-13-2018

Agenda

- Project
- New Features
- Roadmap
- Q & A

About

- Pavol Loffay, Red Hat
  - https://github.com/pavolloffay
- Joe Farro, Uber Technologies
  - https://github.com/tiffon
- Yuri Shkuro, Uber Technologies
  - https://github.com/yurishkuro

Jaeger - /ˈyāgər/, *noun*: hunter

- Inspired by Google's Dapper and OpenZipkin
- Started at Uber in August 2015
- Open sourced in April 2017
- Joined CNCF in Sep 2017 (incubating)
- Applying for graduation

  https://github.com/cncf/toc/pull/171

Jaeger, a Distributed Tracing Platform

trace collection backend

visualization frontend

instrumentation libraries

data mining platform

https://jaegertracing.io

Technology Stack

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

Project & Community

- 7 maintainers, from Uber and Red Hat
- GitHub stats
  - >6,600 stars, >880 forks
  - >580 contributors
    - >220 authors of commits and pull requests
    - >350 issue creators

CLOUD NATIVE
COMPUTING FOUNDATION

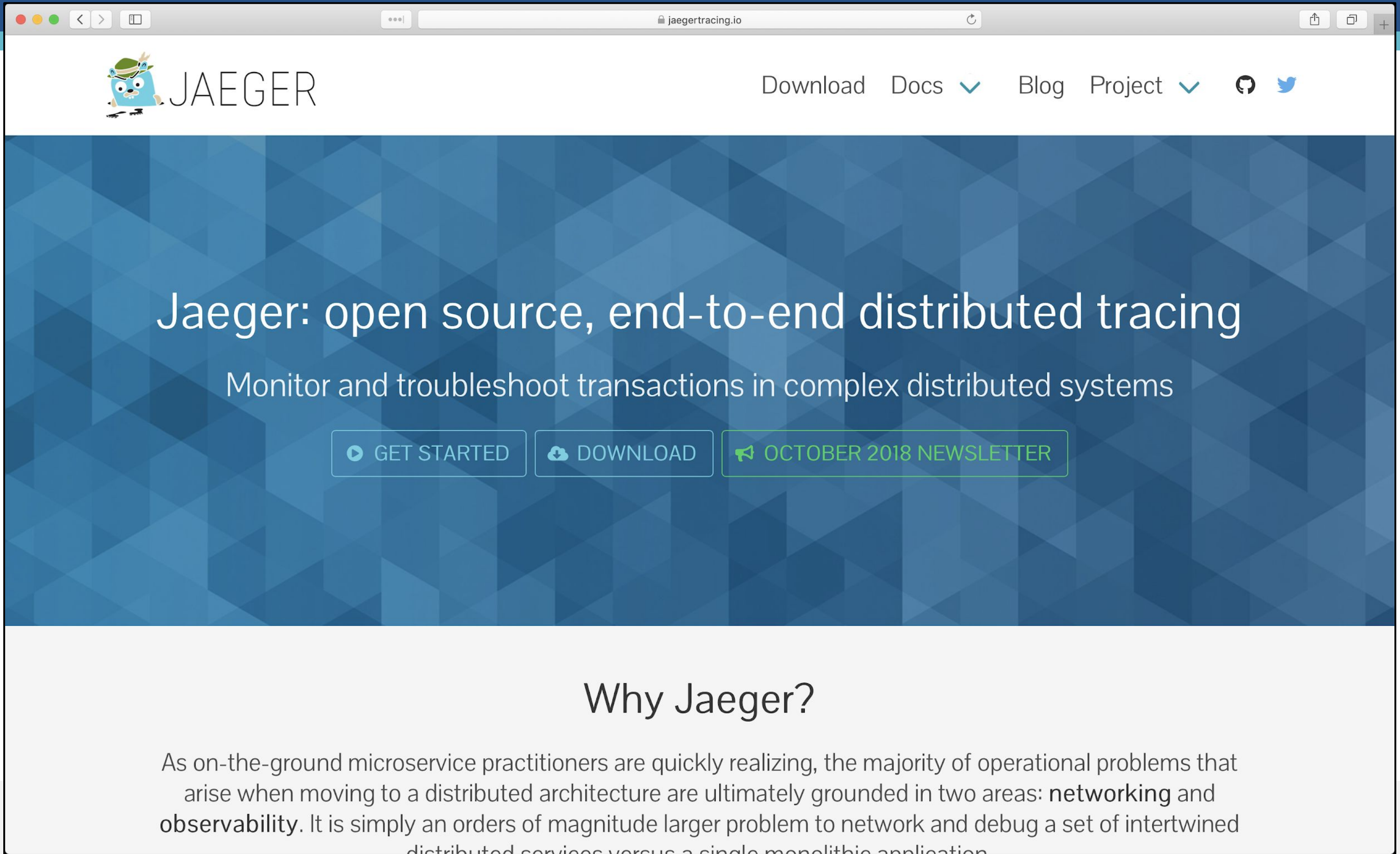
Jaeger 1.8 - 1.9

New Features

New Features

- New website, distributions
- Graph visualizations, trace diffs
- Integrations with other projects
- Async ingestion
- Protobuf & gRPC
- Better Zipkin compatibility

New Website (easy to contribute)



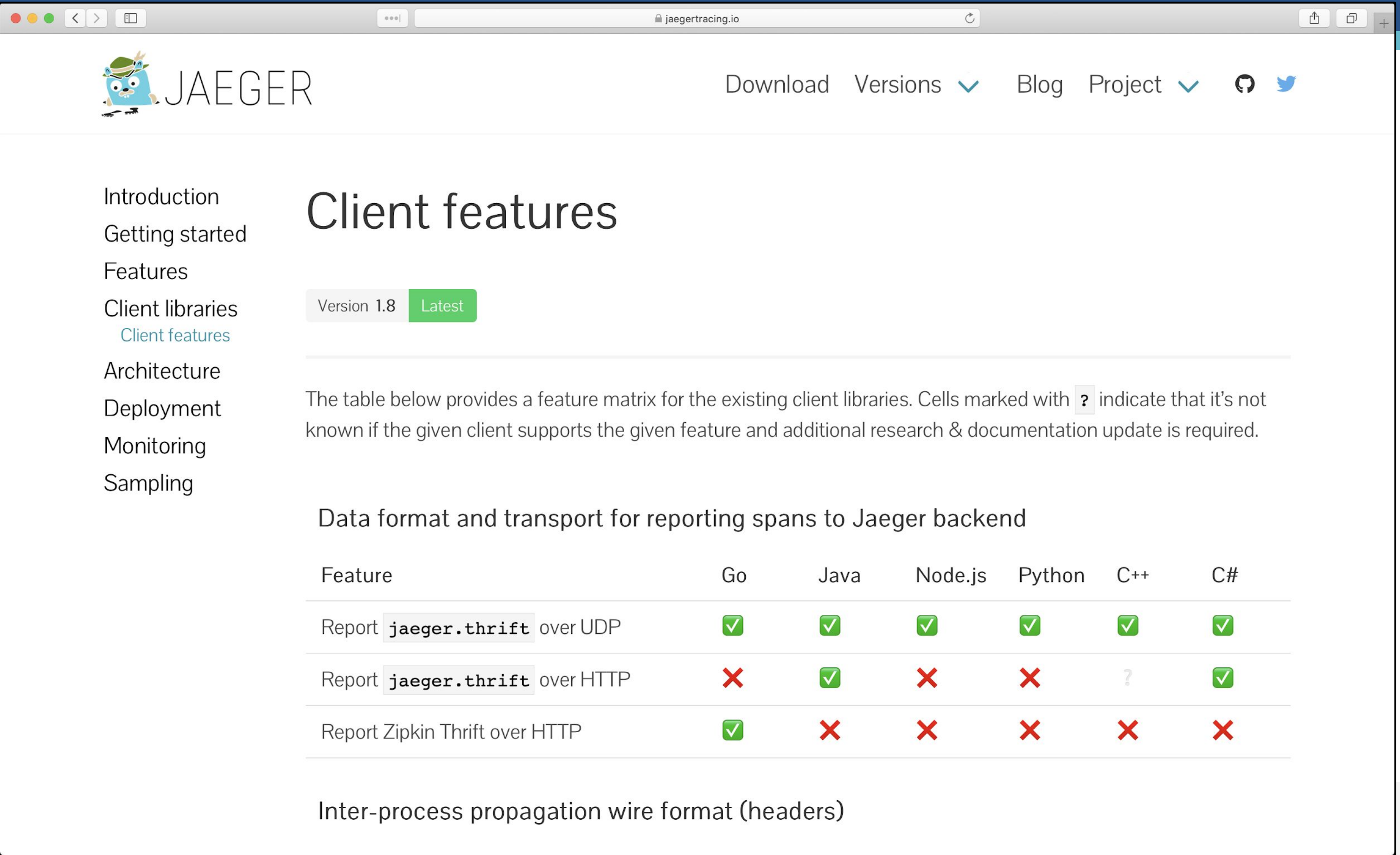
Jaeger: open source, end-to-end distributed tracing

Monitor and troubleshoot transactions in complex distributed systems

Why Jaeger?

As on-the-ground microservice practitioners are quickly realizing, the majority of operational problems that arise when moving to a distributed architecture are ultimately grounded in two areas: networking and observability. It is simply an orders of magnitude larger problem to network and debug a set of intertwined

Example: Client Features matrix (link)



jaegertracing.io

JAEGER

Download Versions Blog Project

Introduction
Getting started
Features
Client libraries
Client features
Architecture
Deployment
Monitoring
Sampling

Client features

Version 1.8 Latest

The table below provides a feature matrix for the existing client libraries. Cells marked with `?` indicate that it's not known if the given client supports the given feature and additional research & documentation update is required.

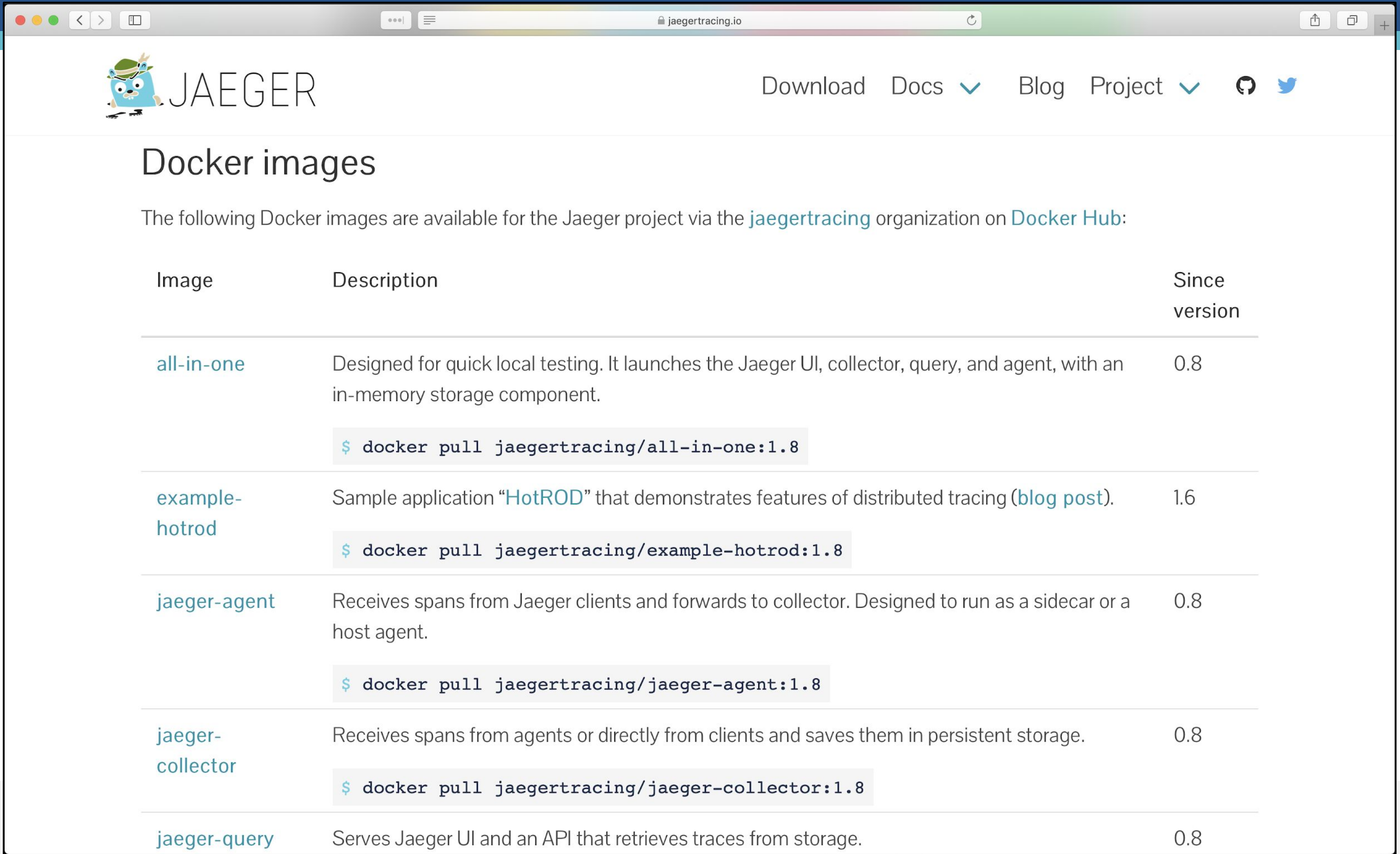
Data format and transport for reporting spans to Jaeger backend

Feature	Go	Java	Node.js	Python	C++	C#
Report `jaeger.thrift` over UDP	✅	✅	✅	✅	✅	✅
Report `jaeger.thrift` over HTTP	❌	✅	❌	❌	?	✅
Report Zipkin Thrift over HTTP	✅	❌	❌	❌	❌	❌

Inter-process propagation wire format (headers)

(link)

Distribution: Docker images

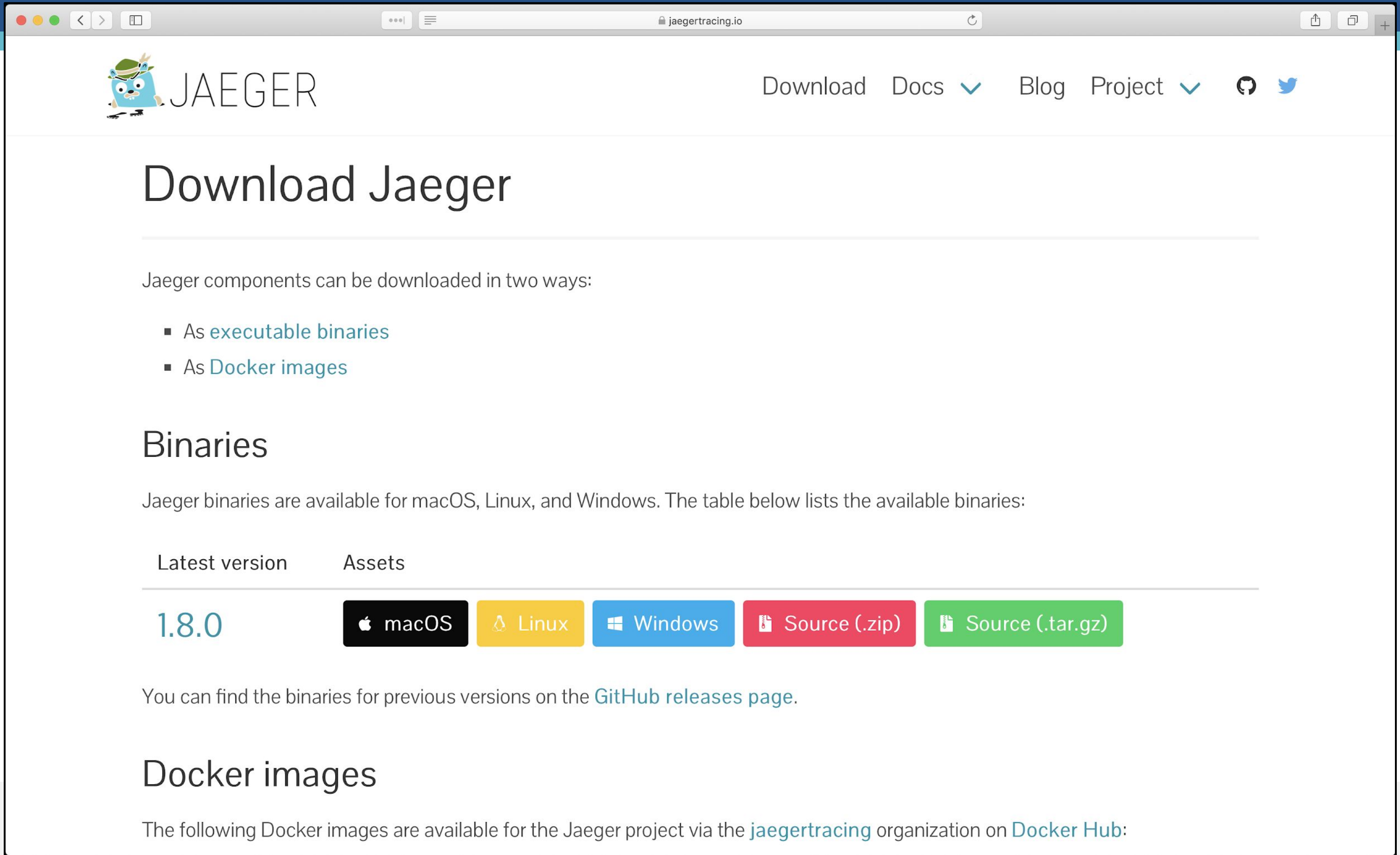


Docker images

The following Docker images are available for the Jaeger project via the jaegertracing organization on Docker Hub:

Image	Description	Since version
all-in-one	Designed for quick local testing. It launches the Jaeger UI, collector, query, and agent, with an in-memory storage component. `$ docker pull jaegertracing/all-in-one:1.8`	0.8
example-hotrod	Sample application "HotROD" that demonstrates features of distributed tracing (blog post). `$ docker pull jaegertracing/example-hotrod:1.8`	1.6
jaeger-agent	Receives spans from Jaeger clients and forwards to collector. Designed to run as a sidecar or a host agent. `$ docker pull jaegertracing/jaeger-agent:1.8`	0.8
jaeger-collector	Receives spans from agents or directly from clients and saves them in persistent storage. `$ docker pull jaegertracing/jaeger-collector:1.8`	0.8
jaeger-query	Serves Jaeger UI and an API that retrieves traces from storage.	0.8

Binaries (Linux, MacOS, Windows)



Download Jaeger

Jaeger components can be downloaded in two ways:

- As executable binaries
- As Docker images

Binaries

Jaeger binaries are available for macOS, Linux, and Windows. The table below lists the available binaries:

Latest version	Assets
1.8.0	macOS, Linux, Windows, Source (.zip), Source (.tar.gz)

You can find the binaries for previous versions on the GitHub releases page.

Docker images

The following Docker images are available for the Jaeger project via the jaegertracing organization on Docker Hub:

CLOUD NATIVE
COMPUTING FOUNDATION

Graph Visualizations

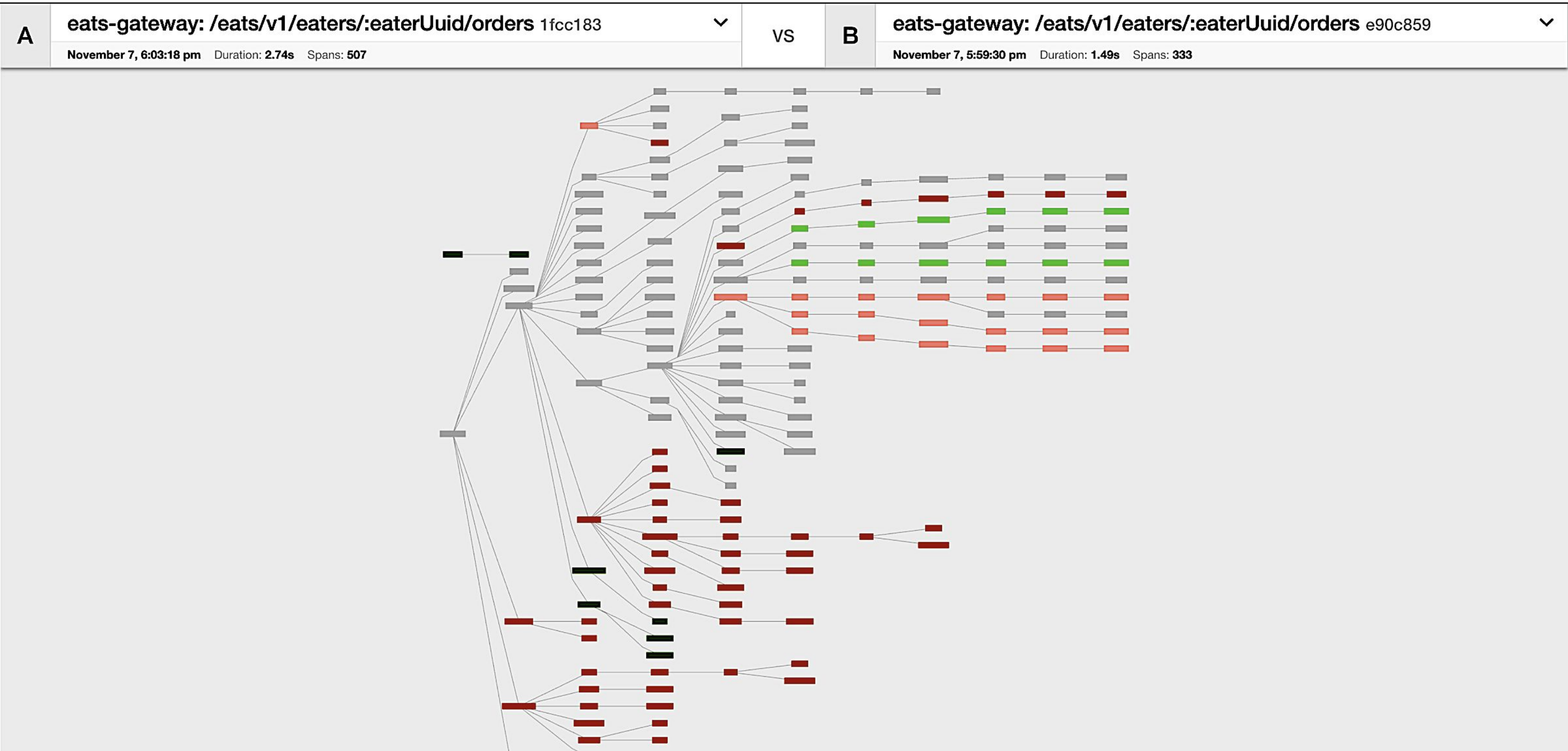
Trade Diffs and Trace Graph

Graph Visualizations

Gantt chart is not great for traces with 10s of thousands of spans

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

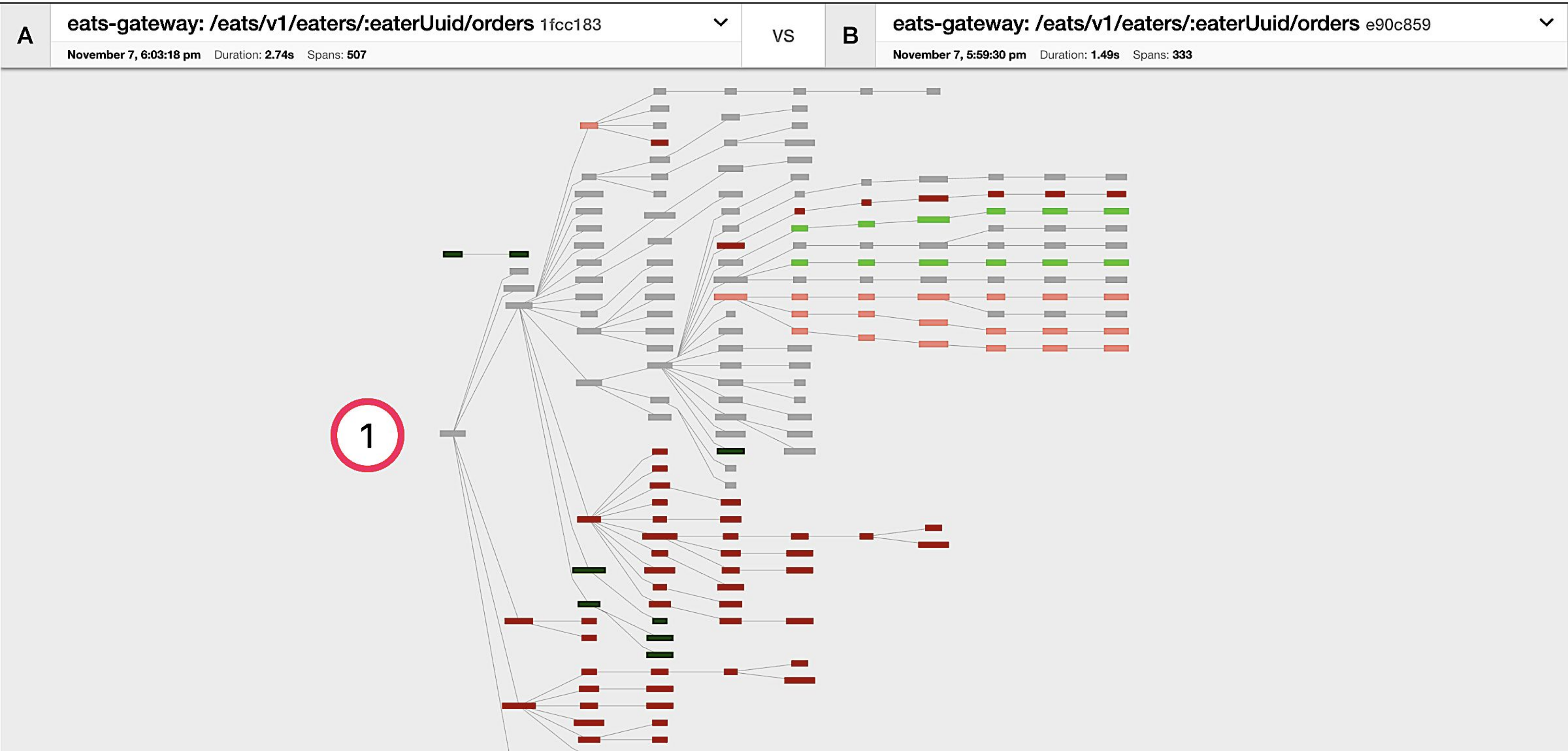
Comparing trace structures – Unified diff


A eats-gateway: /eats/v1/eaters/:eaterUuid/orders 1fcc183
November 7, 6:03:18 pm Duration: 2.74s Spans: 507

VS

B eats-gateway: /eats/v1/eaters/:eaterUuid/orders e90c859
November 7, 5:59:30 pm Duration: 1.49s Spans: 333

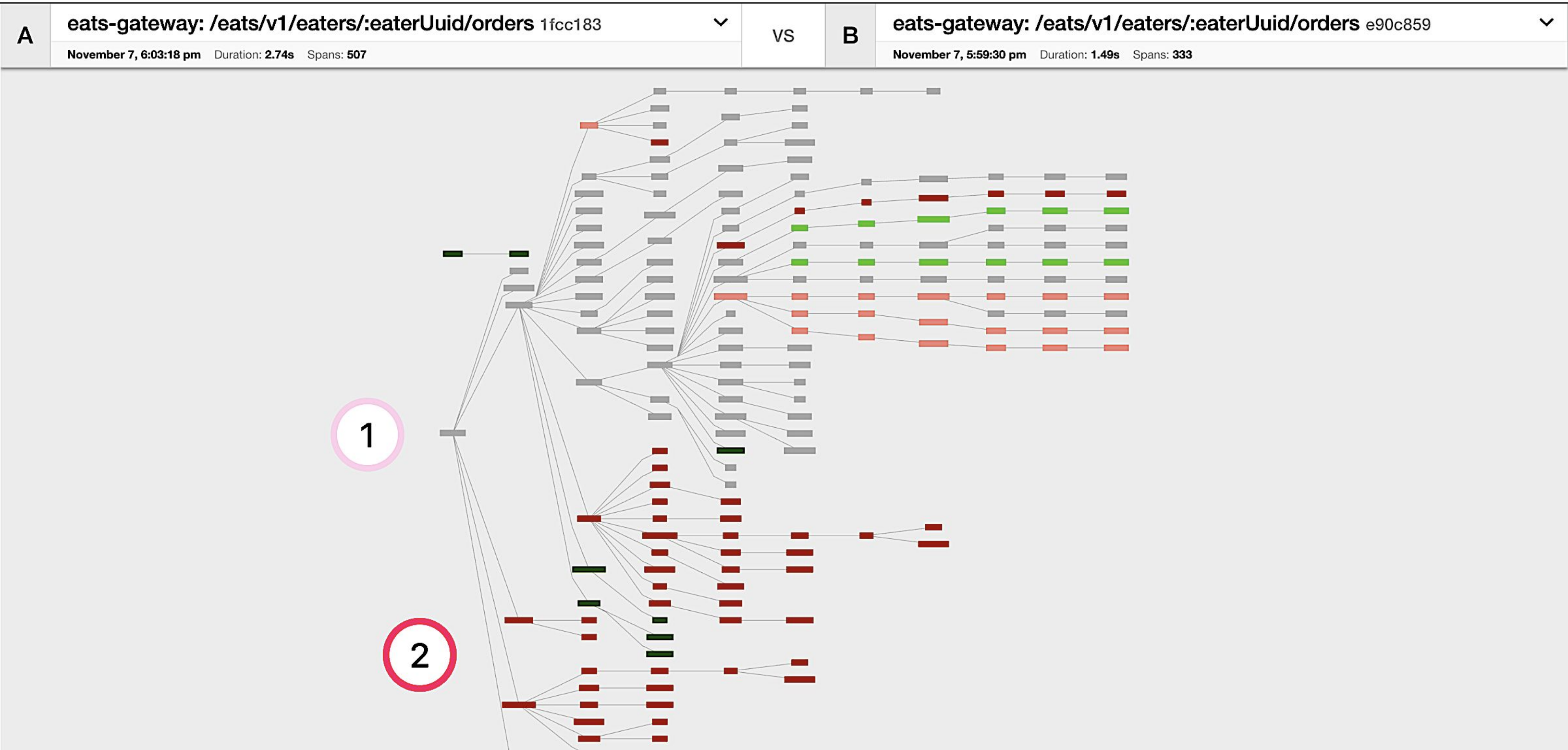
Comparing trace structures – Shared structure


A eats-gateway: /eats/v1/eaters/:eaterUuid/orders 1fcc183
November 7, 6:03:18 pm Duration: 2.74s Spans: 507

VS

B eats-gateway: /eats/v1/eaters/:eaterUuid/orders e90c859
November 7, 5:59:30 pm Duration: 1.49s Spans: 333

Comparing trace structures – Absent in one or the traces

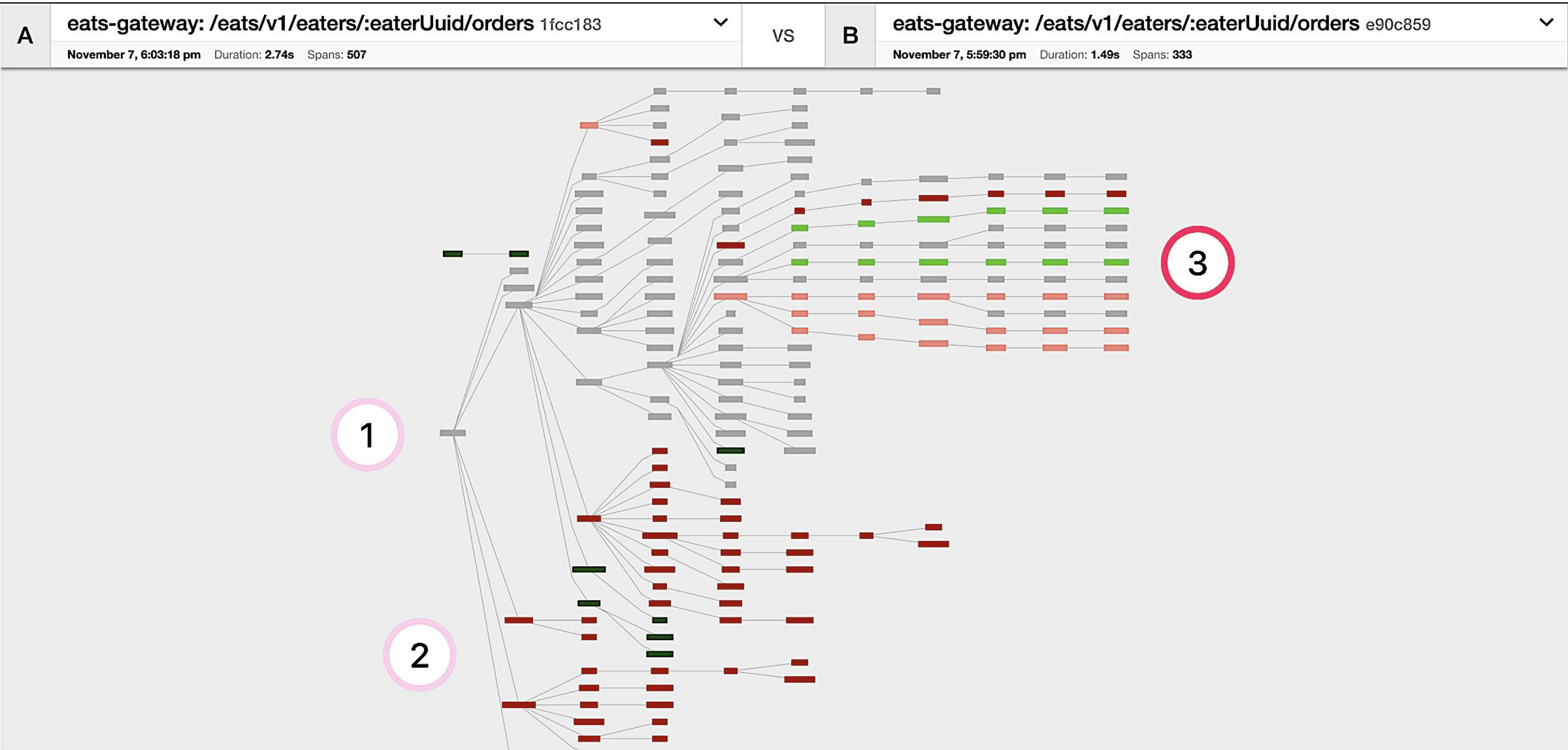


A eats-gateway: /eats/v1/eaters/:eaterUuid/orders 1fcc183
November 7, 6:03:18 pm Duration: 2.74s Spans: 507

VS

B eats-gateway: /eats/v1/eaters/:eaterUuid/orders e90c859
November 7, 5:59:30 pm Duration: 1.49s Spans: 333

Comparing trace structures – More or less within a node

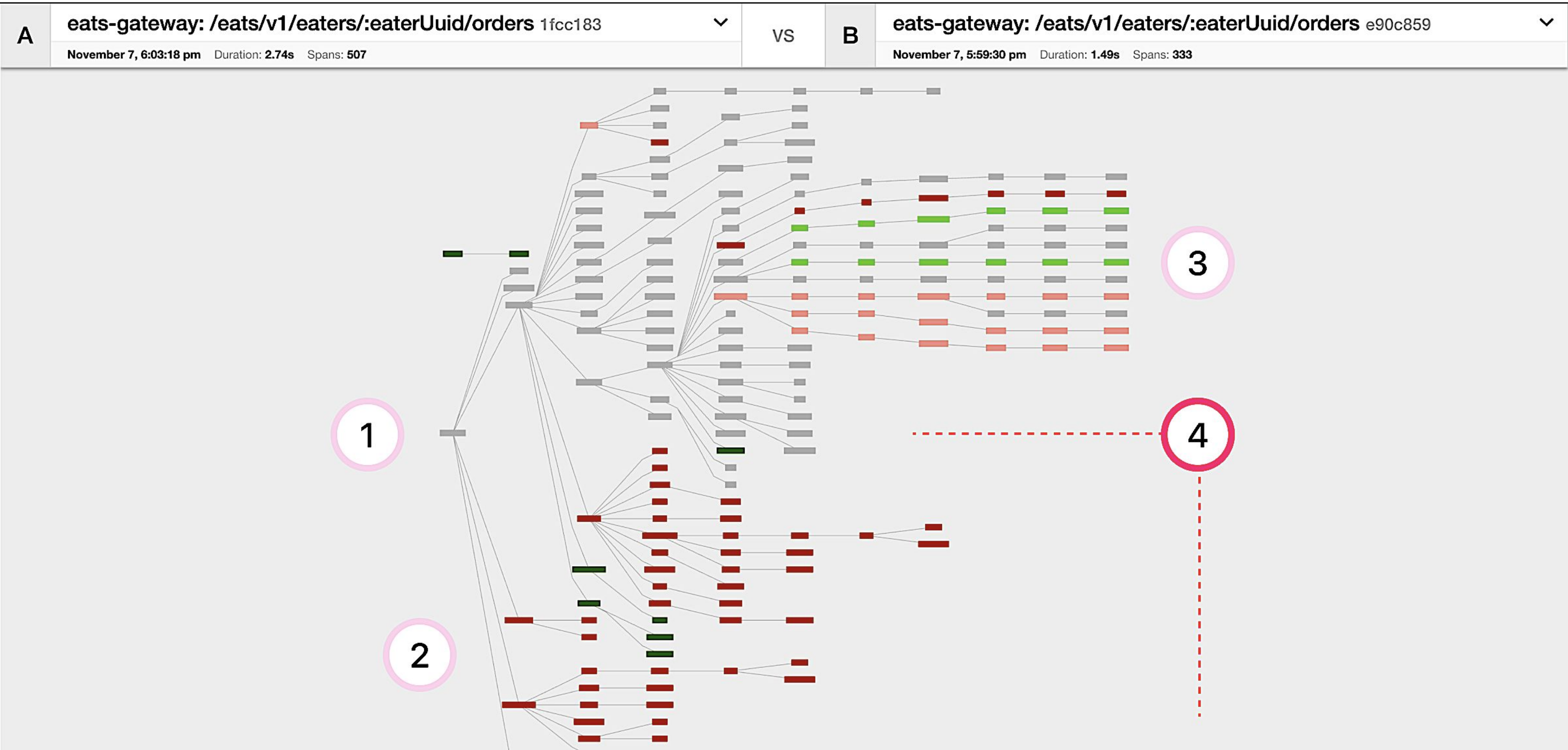


A | eats-gateway: /eats/v1/eaters/:eaterUuid/orders 1fcc183
November 7, 6:03:18 pm Duration: 2.74s Spans: 507

VS

B | eats-gateway: /eats/v1/eaters/:eaterUuid/orders e90c859
November 7, 5:59:30 pm Duration: 1.49s Spans: 333

Comparing trace structures – Substantial divergence

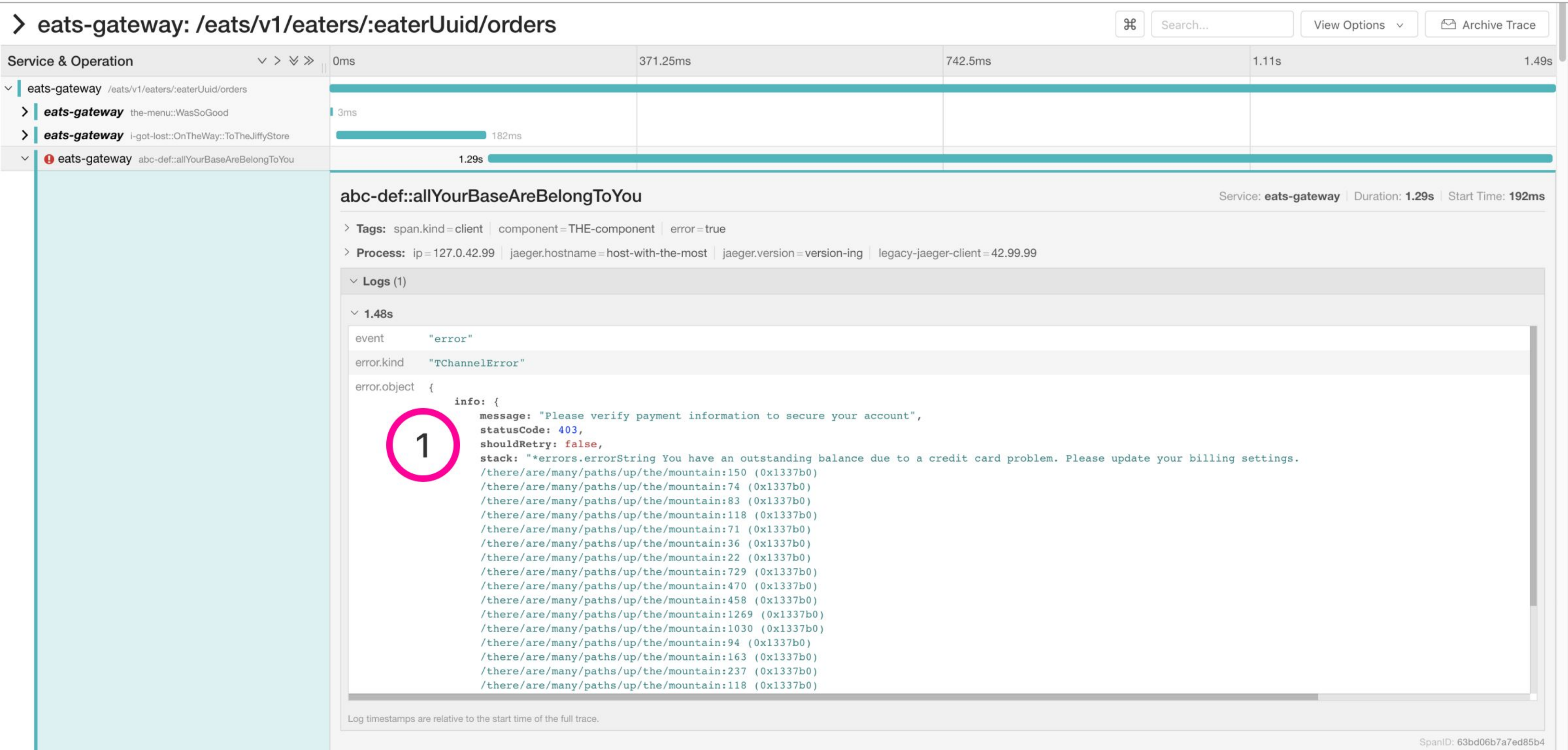


A eats-gateway: /eats/v1/eaters/:eaterUuid/orders 1fcc183
November 7, 6:03:18 pm Duration: 2.74s Spans: 507

VS

B eats-gateway: /eats/v1/eaters/:eaterUuid/orders e90c859
November 7, 5:59:30 pm Duration: 1.49s Spans: 333

"You have an outstanding balance..."



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

eats-gateway /eats/v1/eaters/:eaterUuid/orders
eats-gateway the-menu::WasSoGood 3ms
eats-gateway i-got-lost::OnTheWay::ToTheJiffyStore 182ms
eats-gateway abc-def::allYourBaseAreBelongToYou 1.29s

abc-def::allYourBaseAreBelongToYou

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

Tags: span.kind = client | component = THE-component | error = true

Process: ip = 127.0.42.99 | jaeger.hostname = host-with-the-most | jaeger.version = version-ing | legacy-jaeger-client = 42.99.99

Logs (1)

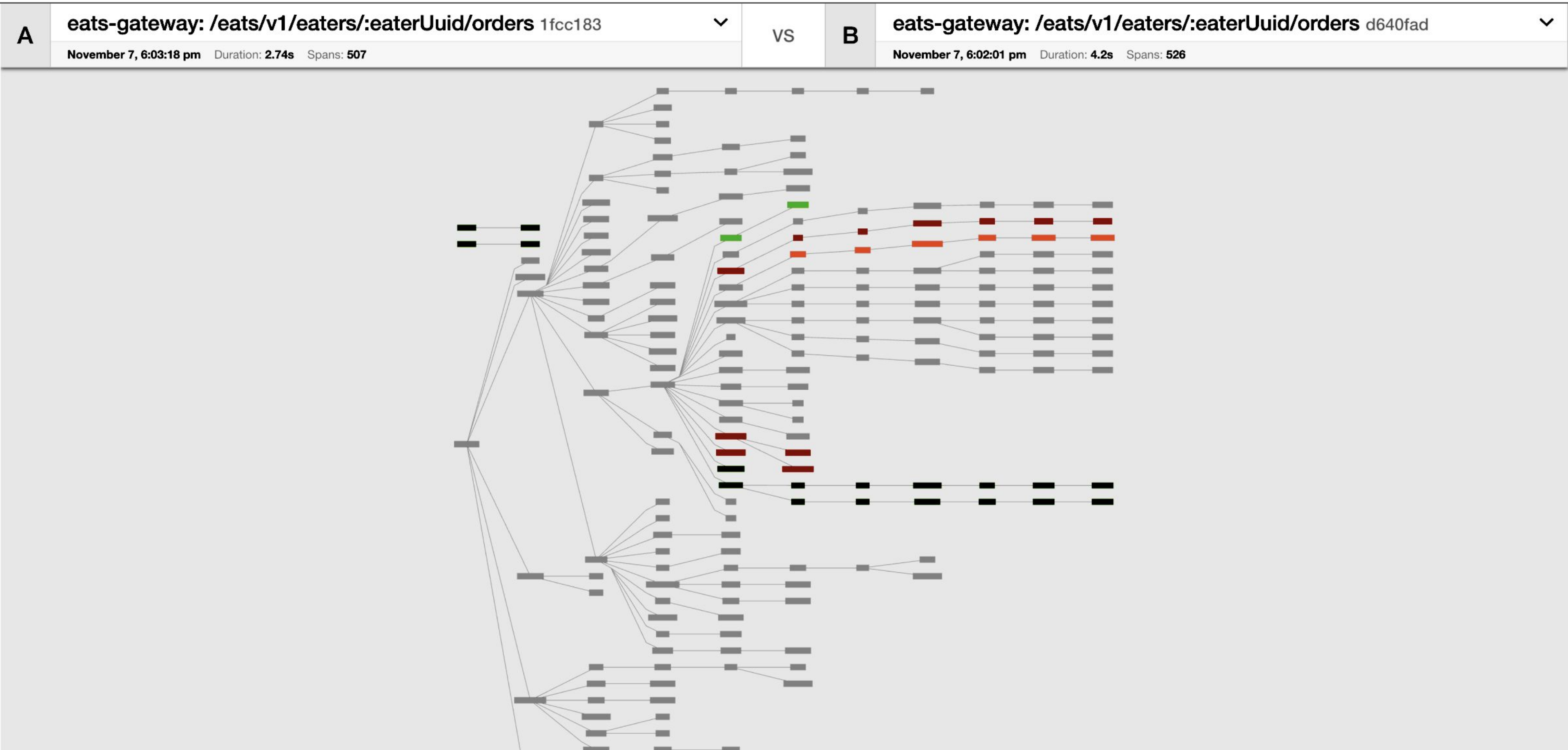
1.48s

```
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/paths/up/the/mountain:150 (0x1337b0)
                       /there/are/many/paths/up/the/mountain:74 (0x1337b0)
                       /there/are/many/paths/up/the/mountain:83 (0x1337b0)
                       /there/are/many/paths/up/the/mountain:118 (0x1337b0)
                       /there/are/many/paths/up/the/mountain:71 (0x1337b0)
                       /there/are/many/paths/up/the/mountain:36 (0x1337b0)
                       /there/are/many/paths/up/the/mountain:22 (0x1337b0)
                       /there/are/many/paths/up/the/mountain:729 (0x1337b0)
                       /there/are/many/paths/up/the/mountain:470 (0x1337b0)
                       /there/are/many/paths/up/the/mountain:458 (0x1337b0)
                       /there/are/many/paths/up/the/mountain:1269 (0x1337b0)
                       /there/are/many/paths/up/the/mountain:1030 (0x1337b0)
                       /there/are/many/paths/up/the/mountain:94 (0x1337b0)
                       /there/are/many/paths/up/the/mountain:163 (0x1337b0)
                       /there/are/many/paths/up/the/mountain:237 (0x1337b0)
                       /there/are/many/paths/up/the/mountain:118 (0x1337b0)
```

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

SpanID: 63bd06b7a7ed85b4

Structural vs. Time

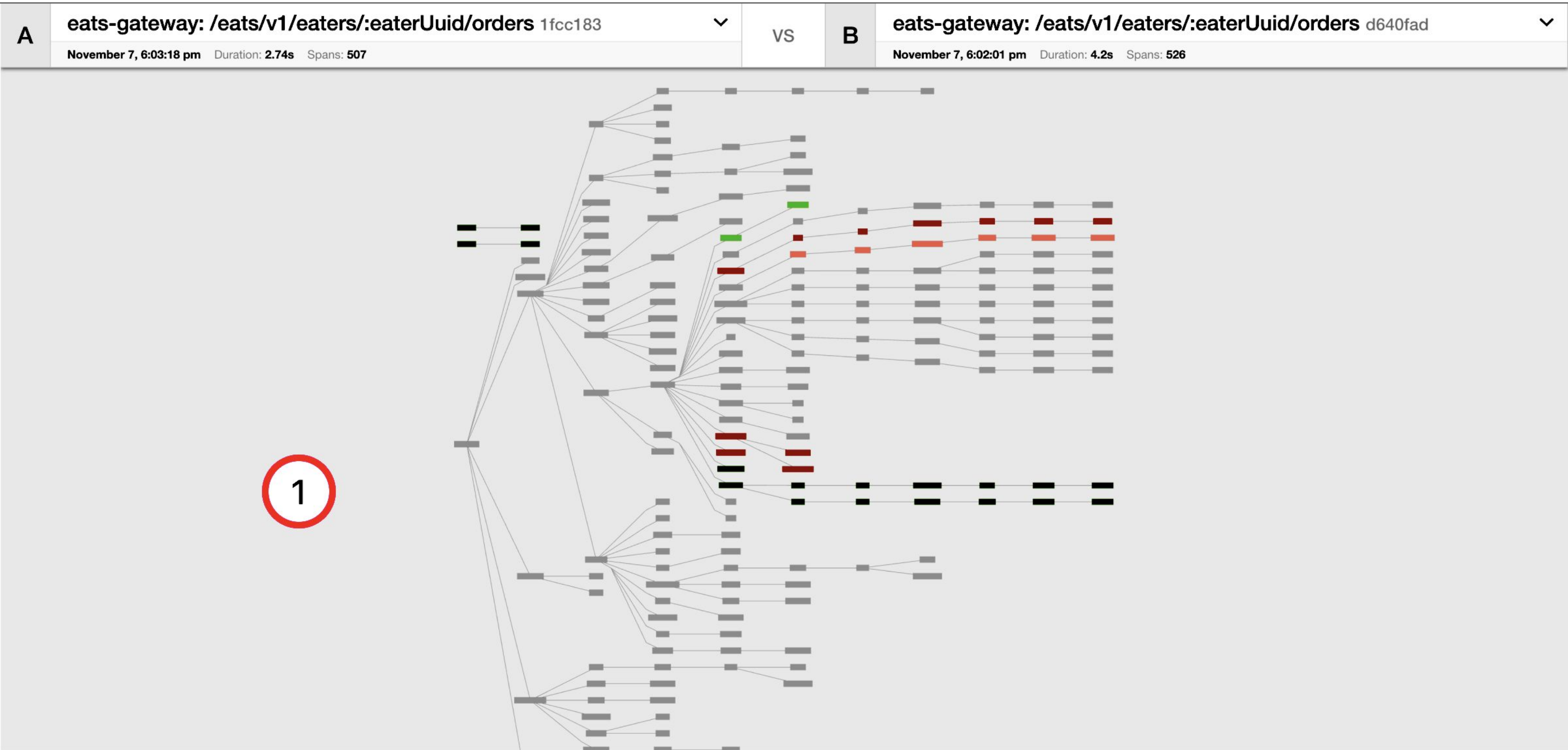


A eats-gateway: /eats/v1/eaters/:eaterUuid/orders 1fcc183
November 7, 6:03:18 pm Duration: 2.74s Spans: 507

VS

B eats-gateway: /eats/v1/eaters/:eaterUuid/orders d640fad
November 7, 6:02:01 pm Duration: 4.2s Spans: 526

Structural vs. Time – Very similar structures

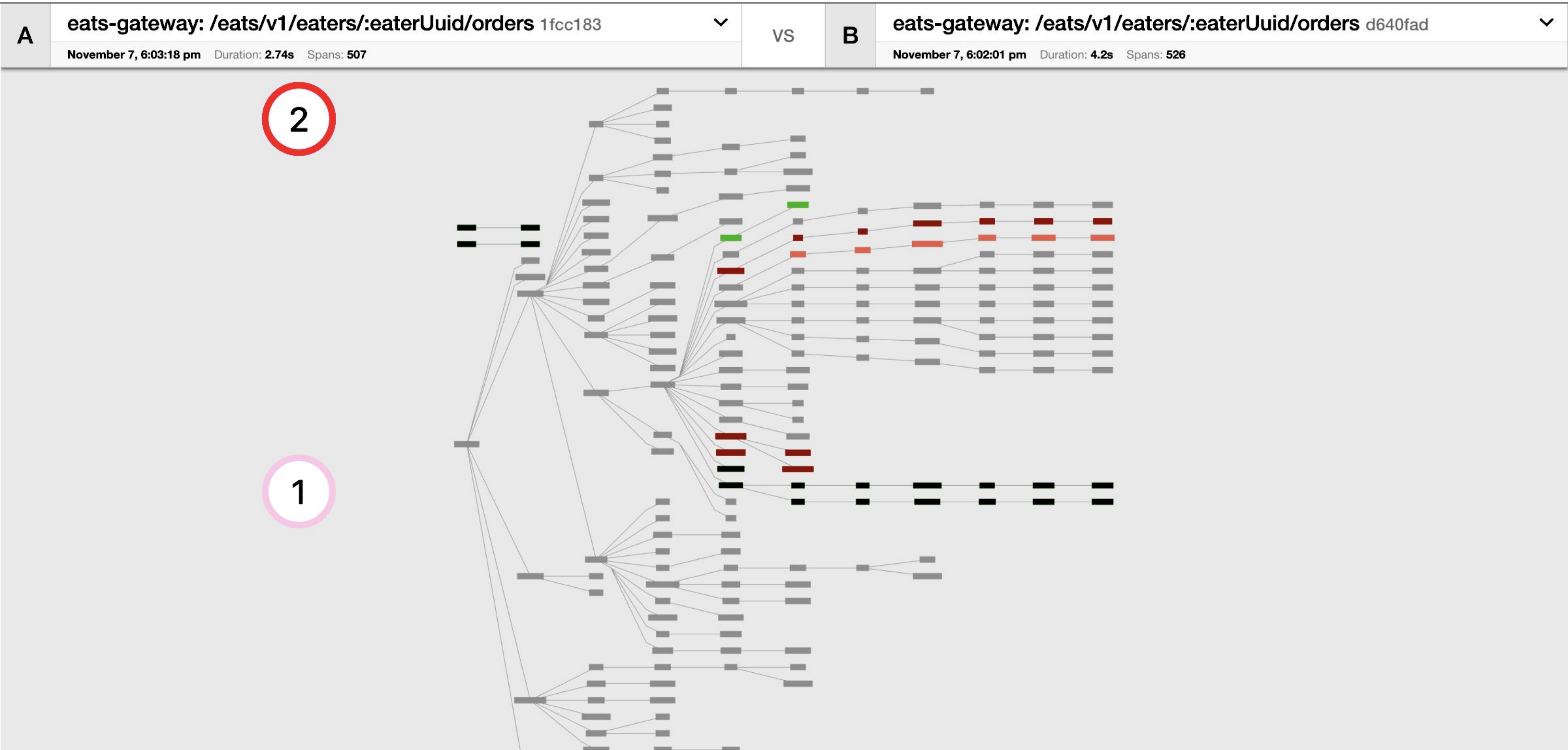


A | eats-gateway: /eats/v1/eaters/:eaterUuid/orders 1fcc183
November 7, 6:03:18 pm Duration: **2.74s** Spans: **507**

VS

B | eats-gateway: /eats/v1/eaters/:eaterUuid/orders d640fad
November 7, 6:02:01 pm Duration: **4.2s** Spans: **526**

Structural vs. Time – 2.74 seconds

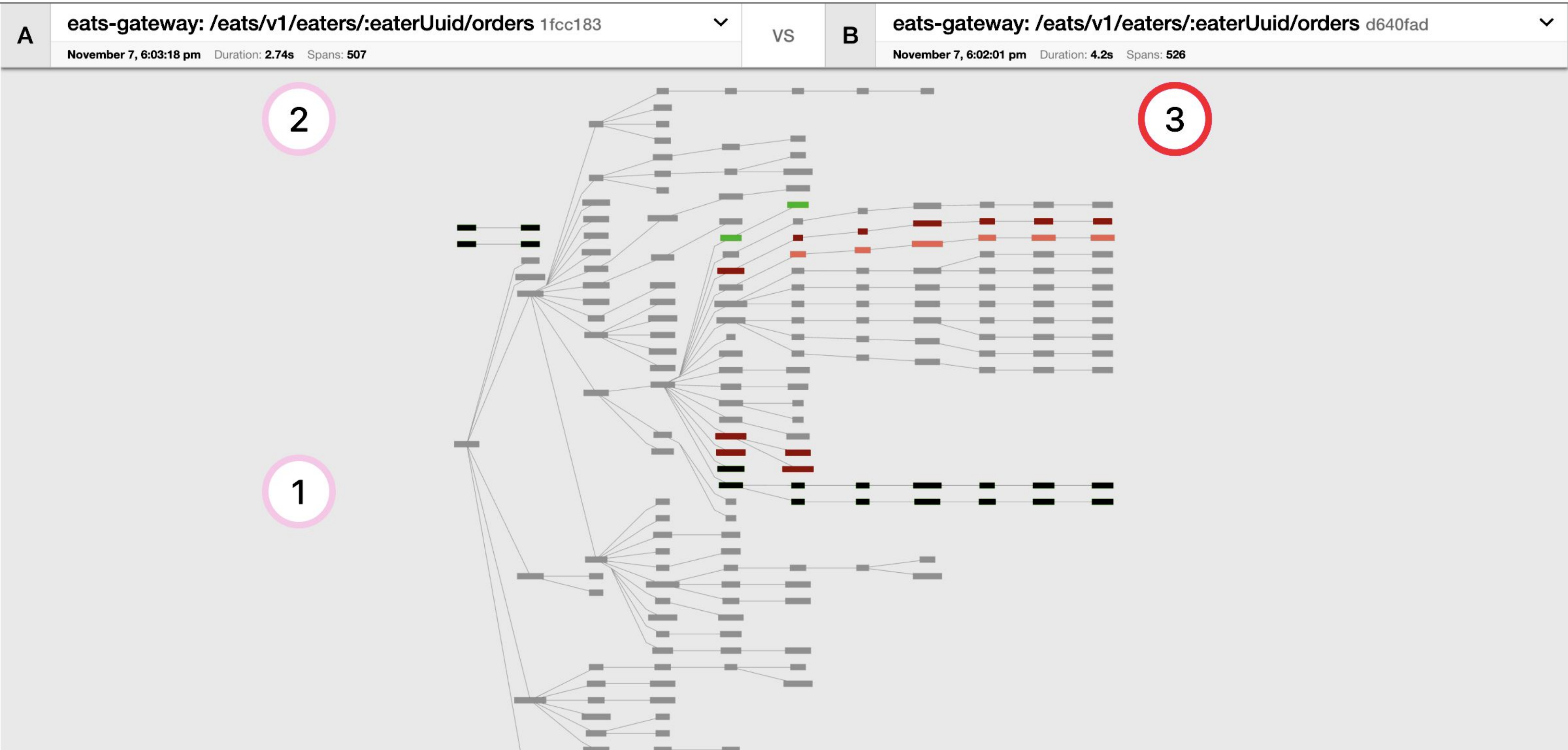


A | eats-gateway: /eats/v1/eaters/:eaterUuid/orders 1fcc183
November 7, 6:03:18 pm Duration: 2.74s Spans: 507

VS

B | eats-gateway: /eats/v1/eaters/:eaterUuid/orders d640fad
November 7, 6:02:01 pm Duration: 4.2s Spans: 526

Structural vs. Time – 50% increase in duration

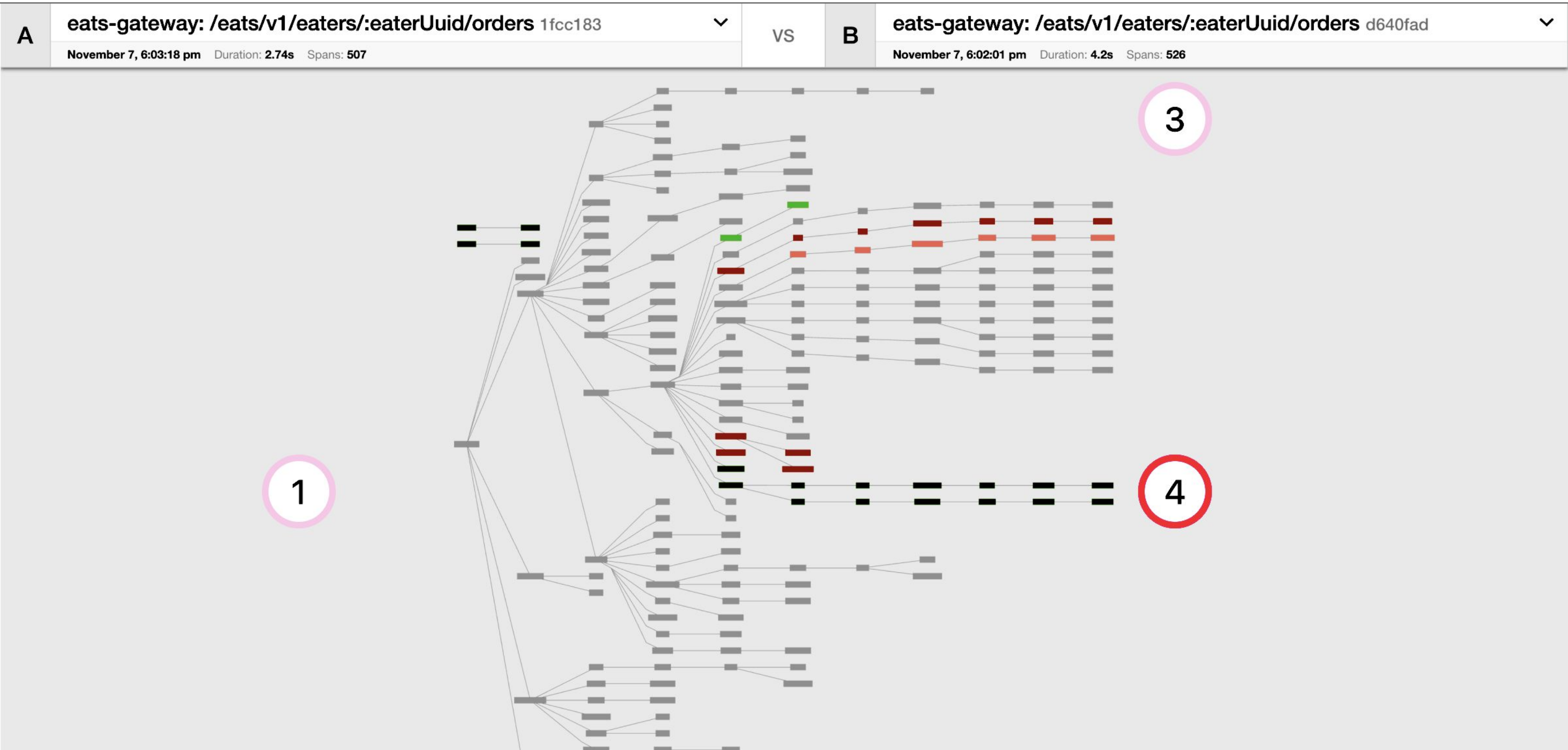


A | eats-gateway: /eats/v1/eaters/:eaterUuid/orders 1fcc183
November 7, 6:03:18 pm Duration: 2.74s Spans: 507

VS

B | eats-gateway: /eats/v1/eaters/:eaterUuid/orders d640fad
November 7, 6:02:01 pm Duration: 4.2s Spans: 526

Structural vs. Time – Are these new spans to blame?



A eats-gateway: /eats/v1/eaters/:eaterUuid/orders 1fcc183

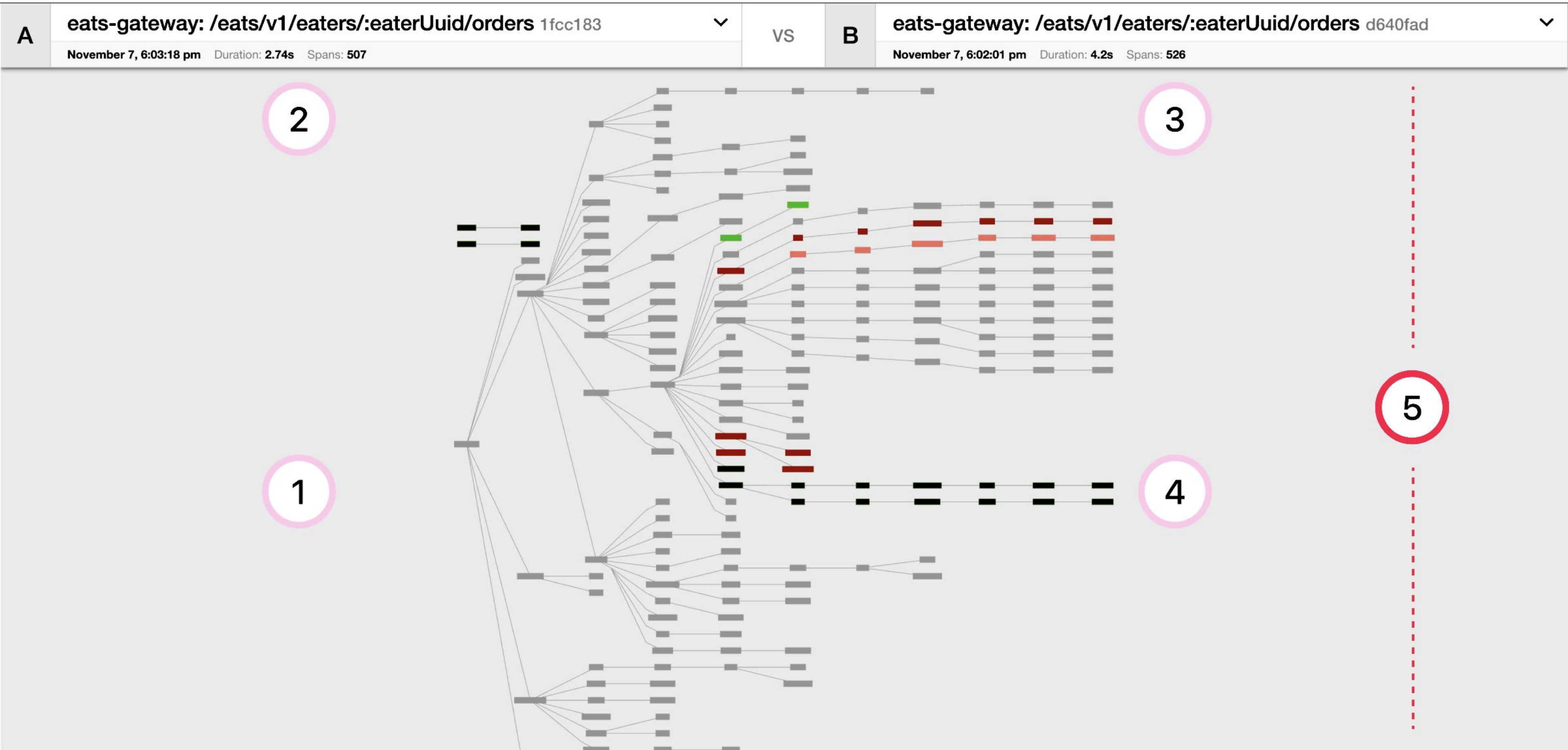
November 7, 6:03:18 pm Duration: 2.74s Spans: 507

VS

B eats-gateway: /eats/v1/eaters/:eaterUuid/orders d640fad

November 7, 6:02:01 pm Duration: 4.2s Spans: 526

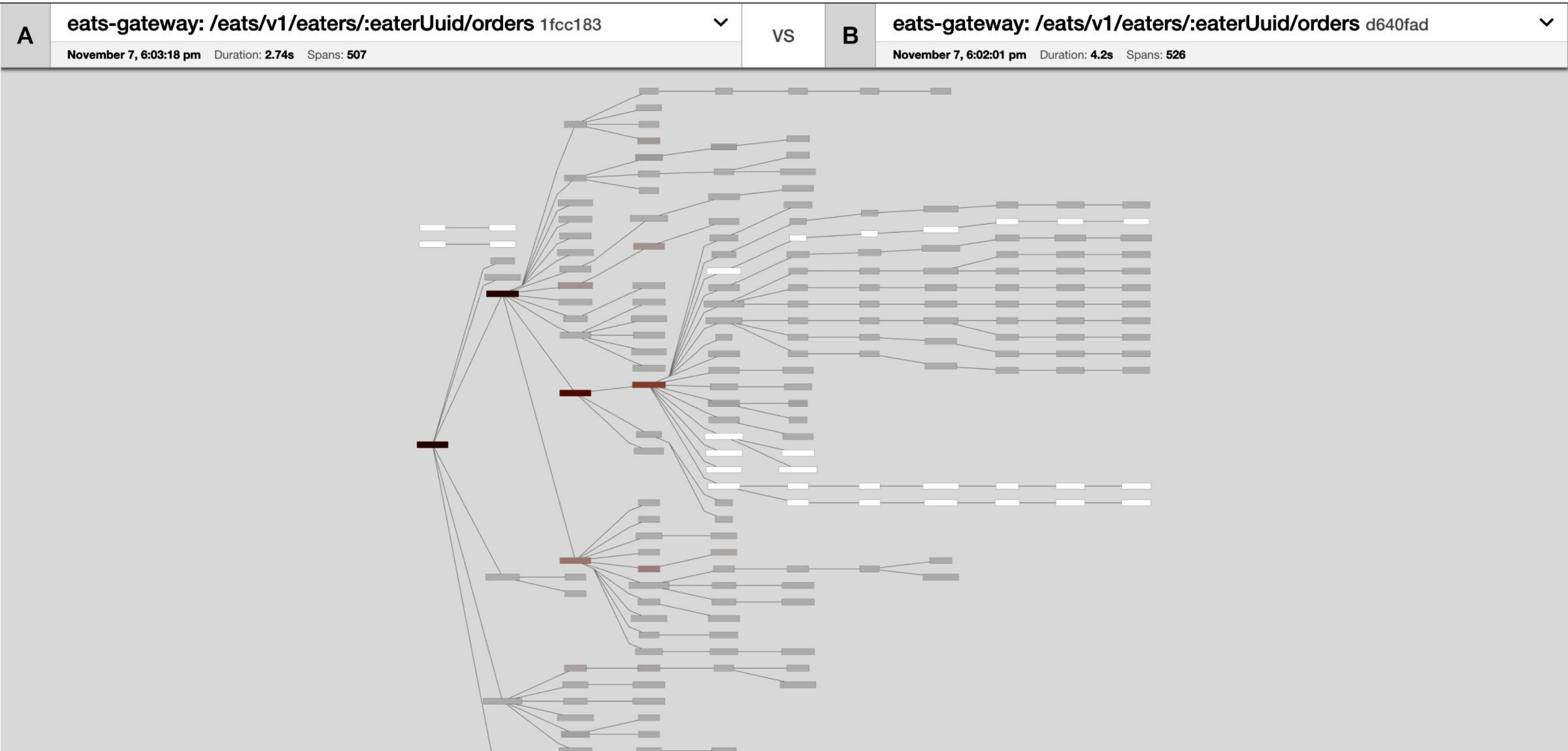
Structural vs. Time – Or is the lag increased throughout?


A eats-gateway: /eats/v1/eaters/:eaterUuid/orders 1fcc183
November 7, 6:03:18 pm Duration: 2.74s Spans: 507

VS

B eats-gateway: /eats/v1/eaters/:eaterUuid/orders d640fad
November 7, 6:02:01 pm Duration: 4.2s Spans: 526

Comparing span durations – Coming soon



A

eats-gateway: /eats/v1/eaters/:eaterUuid/orders 1fcc183

November 7, 6:03:18 pm Duration: **2.74s** Spans: **507**

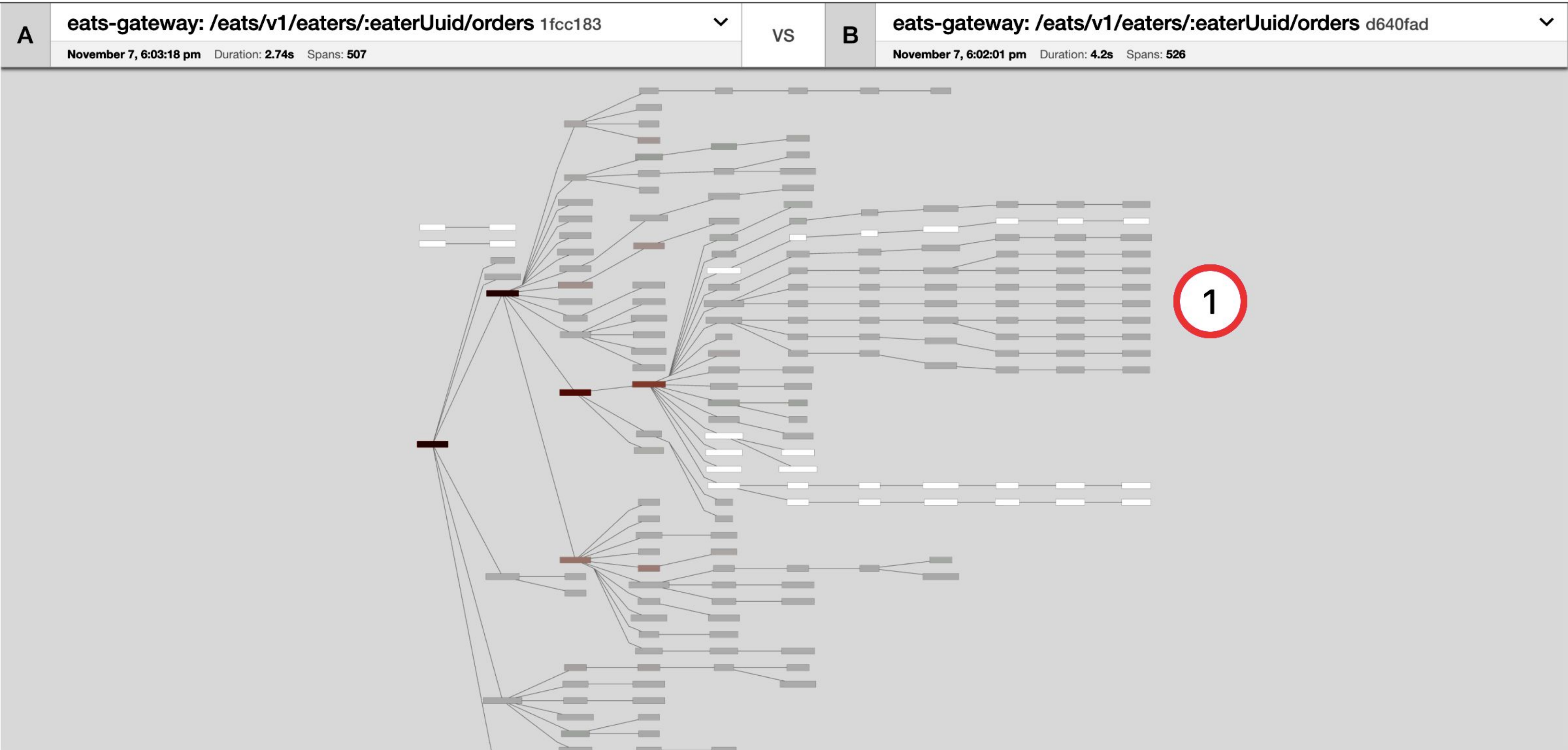
VS

B

eats-gateway: /eats/v1/eaters/:eaterUuid/orders d640fad

November 7, 6:02:01 pm Duration: **4.2s** Spans: **526**

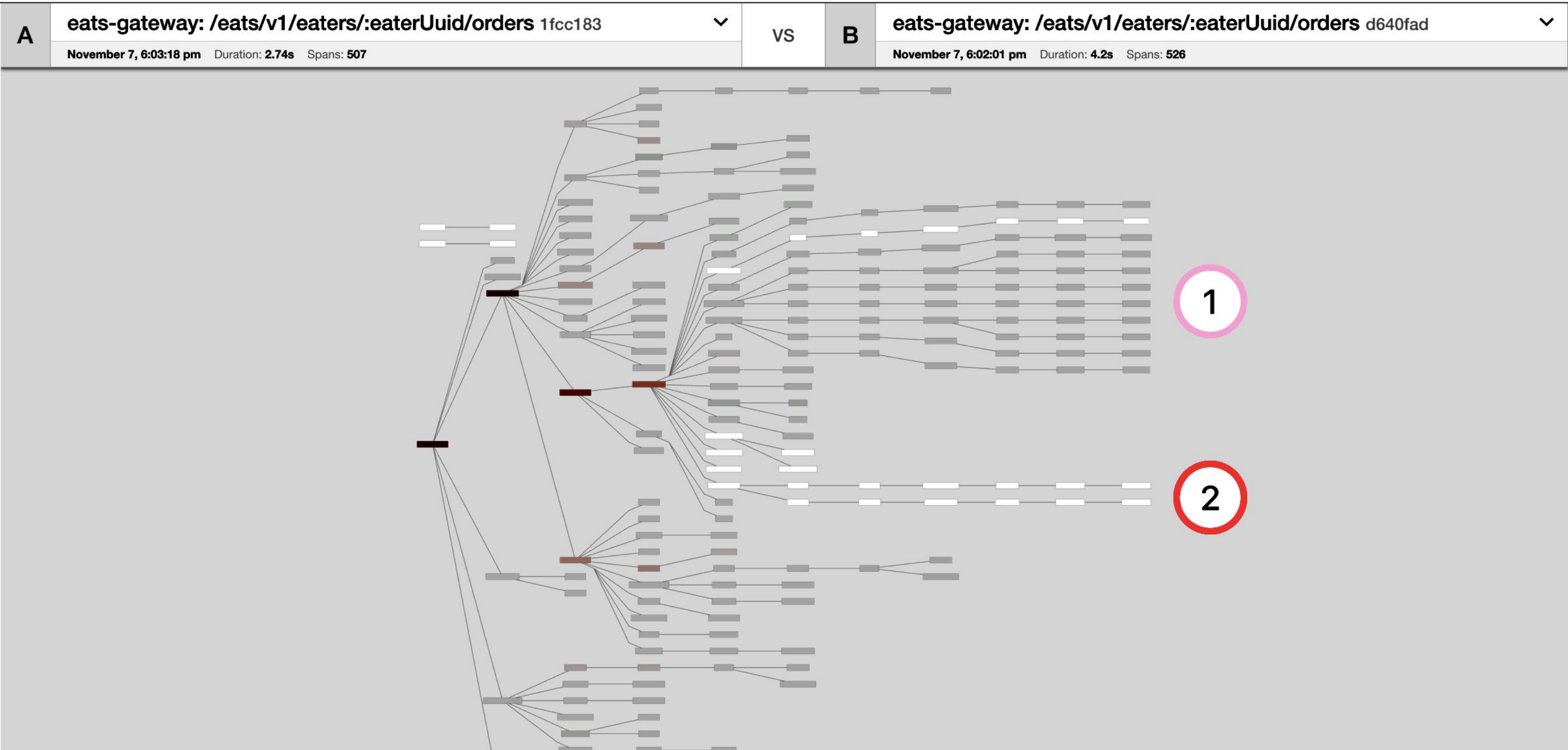
Comparing span durations – Similar durations


A | eats-gateway: /eats/v1/eaters/:eaterUuid/orders 1fcc183
November 7, 6:03:18 pm Duration: 2.74s Spans: 507

VS

B | eats-gateway: /eats/v1/eaters/:eaterUuid/orders d640fad
November 7, 6:02:01 pm Duration: 4.2s Spans: 526

Comparing span durations – Nodes that aren't shared

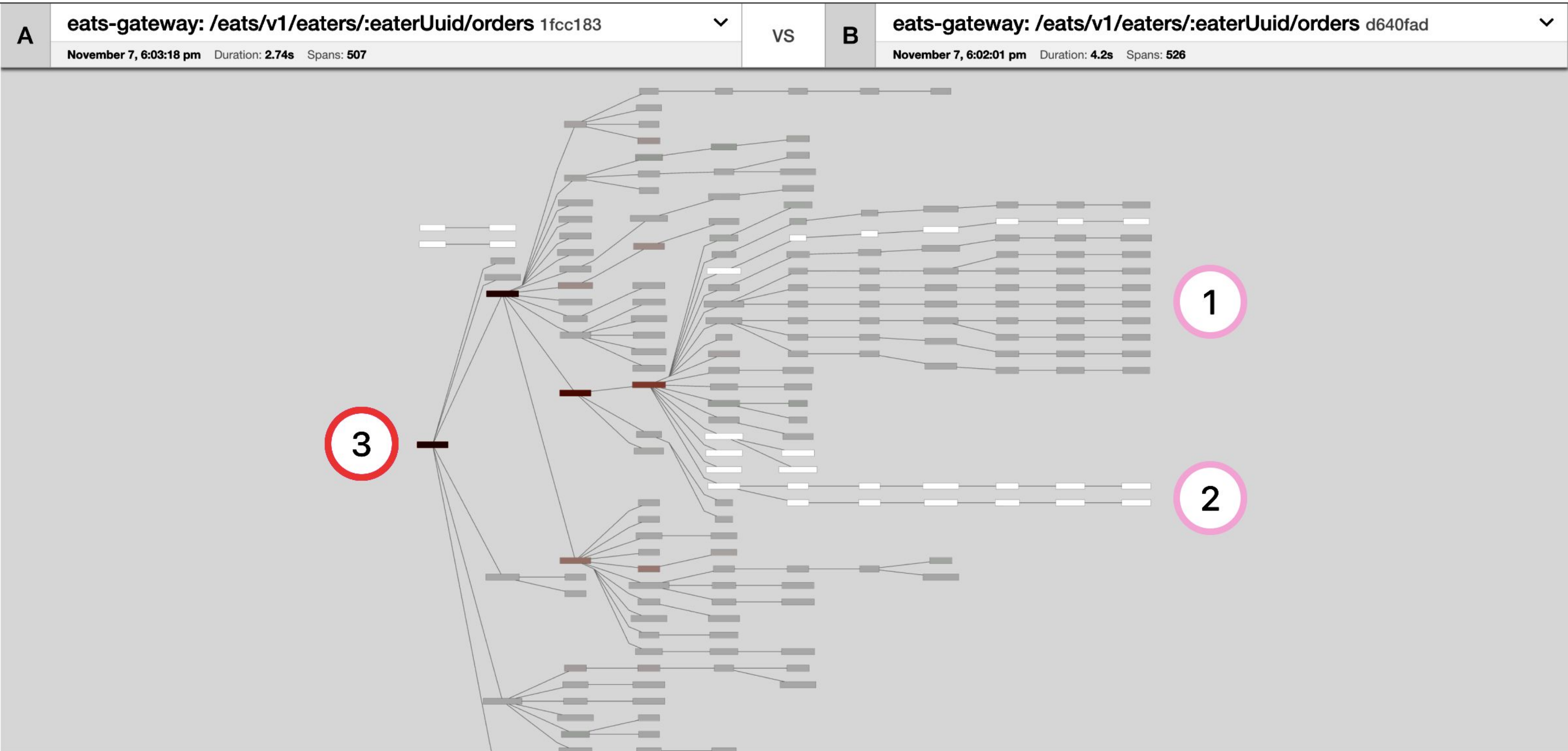


A eats-gateway: /eats/v1/eaters/:eaterUuid/orders 1fcc183
November 7, 6:03:18 pm Duration: 2.74s Spans: 507

VS

B eats-gateway: /eats/v1/eaters/:eaterUuid/orders d640fad
November 7, 6:02:01 pm Duration: 4.2s Spans: 526

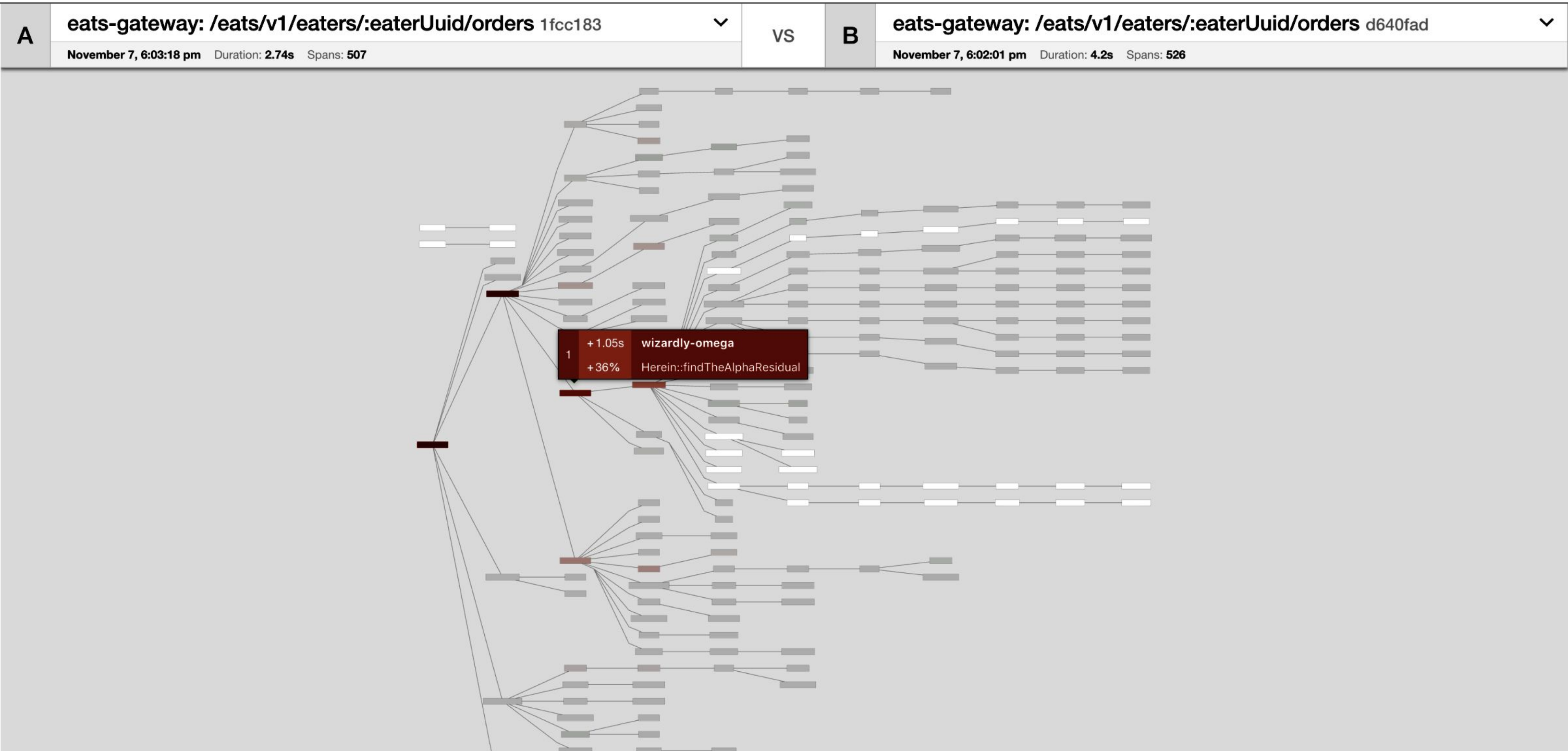
Comparing span durations – Follow the slower nodes


A eats-gateway: /eats/v1/eaters/:eaterUuid/orders 1fcc183
November 7, 6:03:18 pm Duration: 2.74s Spans: 507

VS

B eats-gateway: /eats/v1/eaters/:eaterUuid/orders d640fad
November 7, 6:02:01 pm Duration: 4.2s Spans: 526

Comparing span durations – Coming soon...



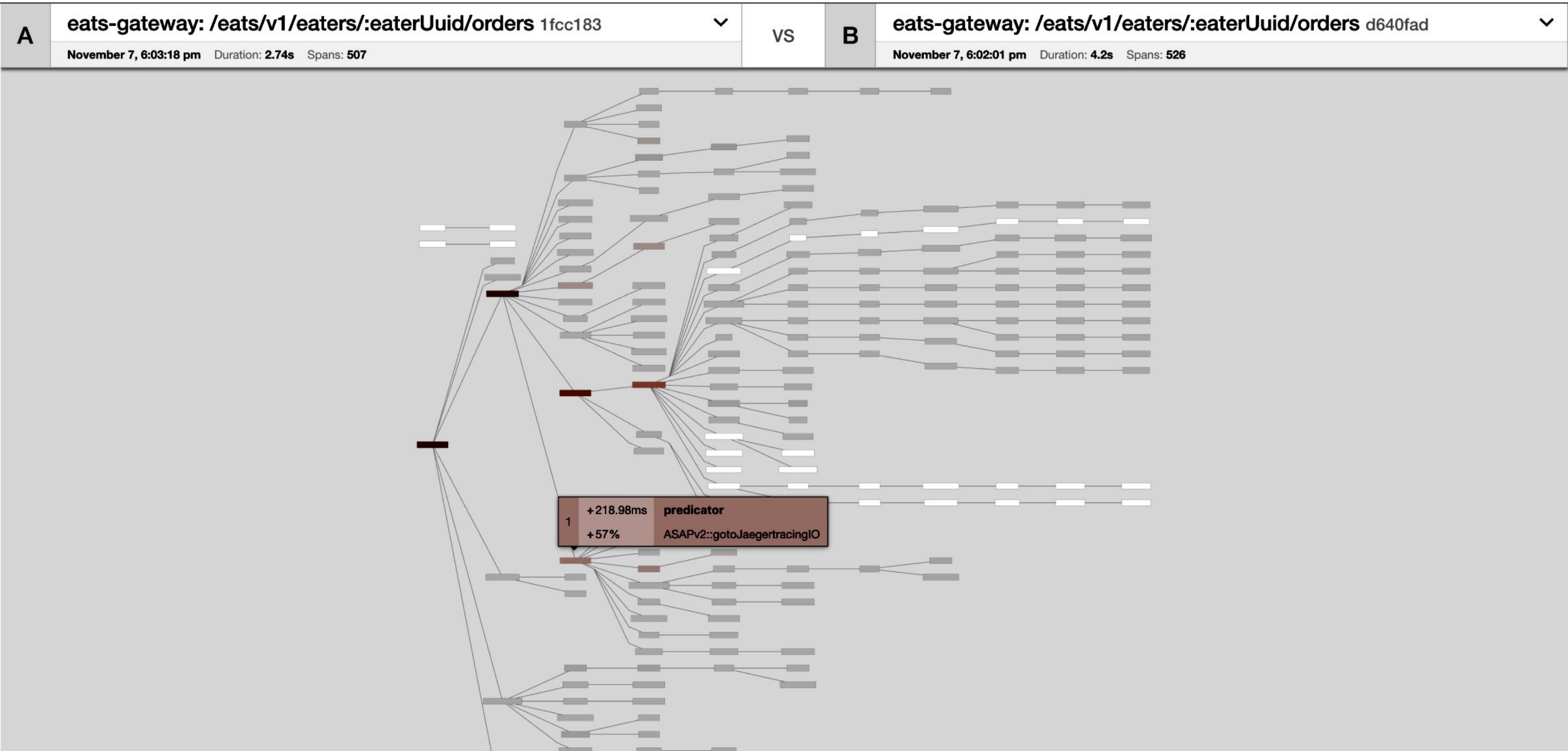
A eats-gateway: /eats/v1/eaters/:eaterUuid/orders 1fcc183
November 7, 6:03:18 pm Duration: 2.74s Spans: 507

VS

B eats-gateway: /eats/v1/eaters/:eaterUuid/orders d640fad
November 7, 6:02:01 pm Duration: 4.2s Spans: 526

1 +1.05s +36% wizardly-omega Herein::findTheAlphaResidual

Comparing span durations – Coming soon...



A | eats-gateway: /eats/v1/eaters/:eaterUuid/orders 1fcc183
November 7, 6:03:18 pm Duration: 2.74s Spans: 507

VS

B | eats-gateway: /eats/v1/eaters/:eaterUuid/orders d640fad
November 7, 6:02:01 pm Duration: 4.2s Spans: 526

1 | +218.98ms | predicator
+57% | ASAPv2::gotoJaegertracingIO

Graph Visualizations

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

CLOUD NATIVE
COMPUTING FOUNDATION

Integrations

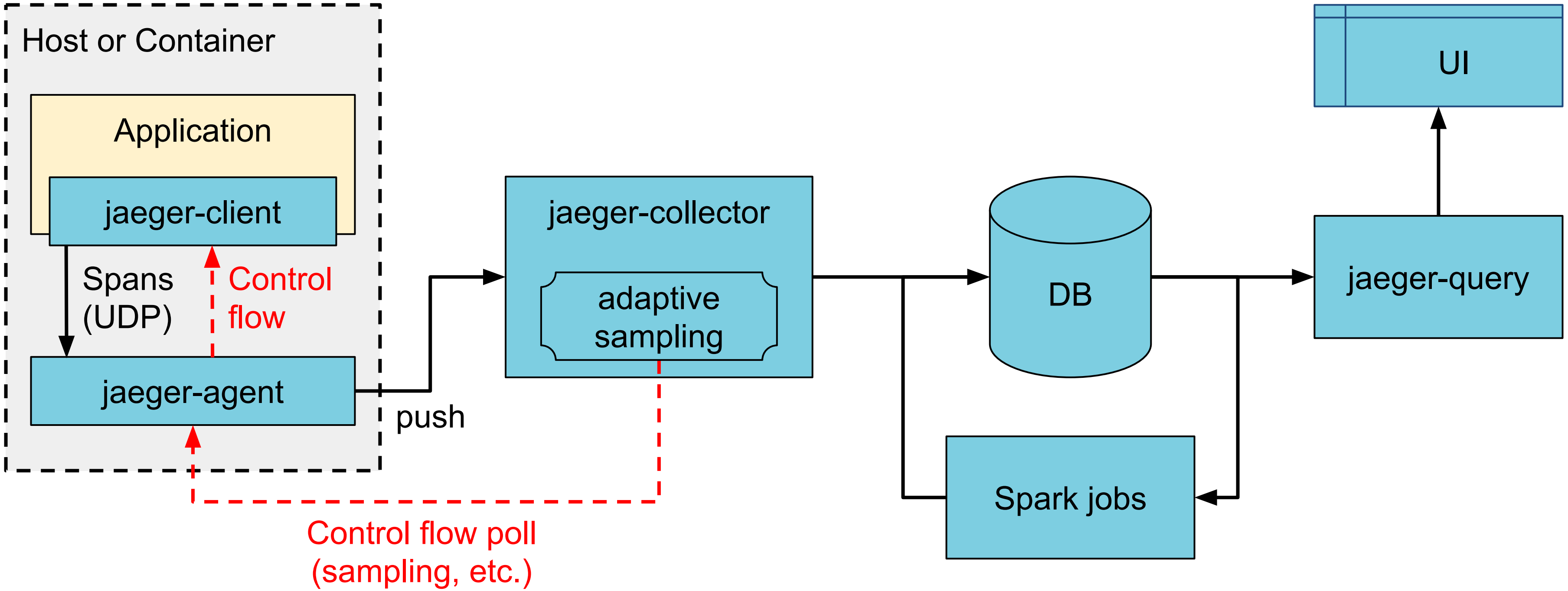
Integrations

- Jaeger Operator for Kubernetes
  - https://github.com/jaegertracing/jaeger-operator
- OpenCensus libraries and agent ship with exporters for Jaeger
  - https://opencensus.io/guides/exporters/supported-exporters/java/jaeger/
- Istio comes with Jaeger included
  - https://istio.io/docs/tasks/telemetry/distributed-tracing/
- Envoy works with Jaeger native C++ client
  - https://www.envoyproxy.io/docs/envoy/latest/start/sandboxes/jaeger_native_tracing
- Eclipse Trace Compass incubator supports importing Jaeger traces
  - https://github.com/tuxology/tracevizlab/tree/master/labs/303-jaeger-opentracing-traces

CLOUD NATIVE
COMPUTING FOUNDATION

Asynchronous Ingestion

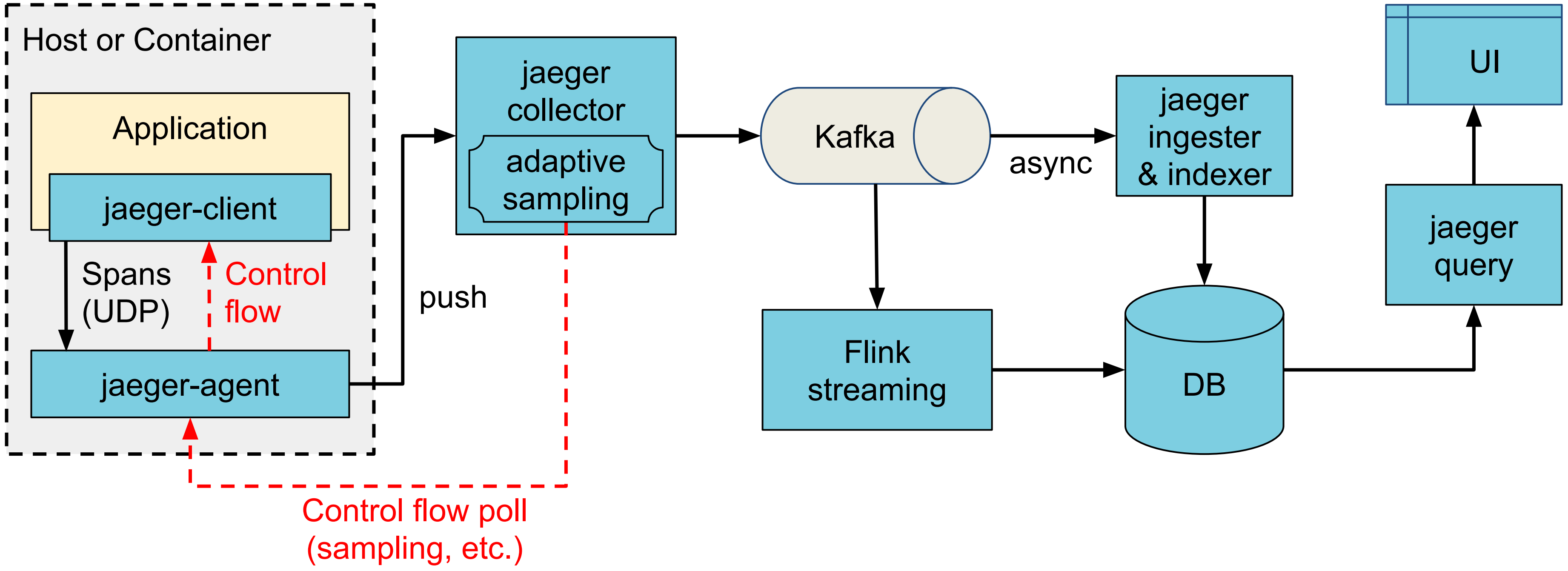
Architecture 2017: Push



Asynchronous span ingestion

- Push model was struggling to keep up with traffic spikes
  - Because of sync storage writes
  - Collectors had to drop data randomly
- Kafka is much more elastic for writes
  - Just raw bytes, no schema, no indexing
  - A lot less overhead on the write path
- Data in Kafka allows for streaming data mining & aggregations
- Two new components: `jaeger-ingester` and `jaeger-indexer`

Architecture now: Push+Async+Streaming



CLOUD NATIVE
COMPUTING FOUNDATION

Protobuf & gRPC

Enabling roadmap

Protobuf & gRPC

- Internal data model generated from Protobuf IDL
- gRPC connection between `jaeger-agent` and `jaeger-collector`

Why

- gRPC plays better with modern routing than TChannel
- Path to official data model and collector/query APIs
- Protobuf-based JSON API
- Unblock development of storage plugins
- (Thrift still supported for backwards compatibility)

CLOUD NATIVE
COMPUTING FOUNDATION

Zipkin Compatibility

Zipkin Compatibility

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

CLOUD NATIVE
COMPUTING FOUNDATION

Roadmap

http://bit.do/jaeger-roadmap

Adaptive Sampling

Problem

- APIs have endpoints with different QPS
- Service owners do not know the full impact of sampling probability

Adaptive Sampling is per service + endpoint, decided by Jaeger backend based on traffic

Adaptive Sampling Status

- Jaeger clients support per service/endpoint sampling strategies
- Can be statically configured in collector
- Pull requests for dynamic recalculations

Data Pipeline

- Based on Kafka and Apache Flink
- Support aggregations and data mining
- Examples:
    - Pairwise dependencies diagram
    - Path-based dependencies diagram
    - Latency histograms

Storage plugins

- Based on gRPC/Protobuf work
- PRs in progress for proof of concept
- Community support for different storage backends

Partial Spans (community driven)

- Add ability to store/retrieve partial spans
- Use case:
    - Certain workflows are hours long. Unfortunately spans are only emitted once after it's Finished(). "Root span" is missing until the complete workflow is finished.

CLOUD NATIVE
COMPUTING FOUNDATION

Learn More

Website: jaegertracing.io/

Blog: medium.com/jaegertracing

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

CLOUD NATIVE
COMPUTING FOUNDATION

Q & A

Open Discussion