redhat

INTRODUCTION TO OPENTRACING

Tracing Summit 2018

2018-10-25

JURACI PAIXÃO KRÖHLING

Software Engineer

Kiali Project, Distributed Tracing team

TRACING AND PROFILING

OPERATING SYSTEM

STRACE, FTRACE, PERF, ...

VIRTUAL MACHINE

JAVA FLIGHT RECORDER, ...

APPLICATION

TYPICAL APM SOLUTIONS, OPENTRACING, ...

WHAT IS OPENTRACING?

AIMS TO STANDARDIZE THE SEMANTICS FOR **DISTRIBUTED TRACING**, WHICH IS ESPECIALLY USEFUL IN **OBSERVING** DISTRIBUTED SYSTEMS, SUCH AS **MICROSERVICES**.

MICROSERVICES

THEY ARE JUST DISTRIBUTED SYSTEMS, BUT IN THE HUNDREDS (THOUSANDS!)

DISTRIBUTED SYSTEMS FAIL. ALL THE TIME.

MULTIPLE VERSIONS PER SERVICE

A/B TESTS, CANARY RELEASES, ROLLING DEPLOYMENT, ...

MICROSERVICES

CHAOS

MICROSERVICES

CHAOS

YAY! AS LONG AS IT'S OBSERVABLE.

WITH CHAOS, WE GET RESILIENCY.

OBSERVABILITY

WHAT WENT WRONG, WHERE AND WHY

METRICS, LOGS, TRACING

OBSERVABILITY

by @peterbourgon

source: https://peter.bourgon.org/blog/2017/02/21/metrics-tracing-and-logging.html

DISTRIBUTED TRACING

STORY OF A REQUEST ACROSS SERVICES

WHICH SERVICES WERE TOUCHED, WHEN, IN WHICH ORDER, WITH INSTANCE INFORMATION (ROUTING INFO, VERSION, TAGS, ...)

DISTRIBUTED TRACING

MEASURES UNITS OF WORK

STORES IN A DATA STRUCTURE CALLED "SPAN"

REFERENCES OTHER SPANS

CAUSALITY!

CONTEXT PROPAGATION

MOST OF IT IS SIMILAR TO CORRELATION ID

DISTRIBUTED TRACING

OUR CODE

```
chargeCreditCard();
changeOrderStatus();
dispatchEventToInventory();
```

DISTRIBUTED TRACING

OUR CODE WITH DISTRIBUTED TRACING

```
try (Scope scope = tracer.buildSpan("submitOrder").startActive(true)) {

    chargeCreditCard();
    changeOrderStatus();
    dispatchEventToInventory();
}
```

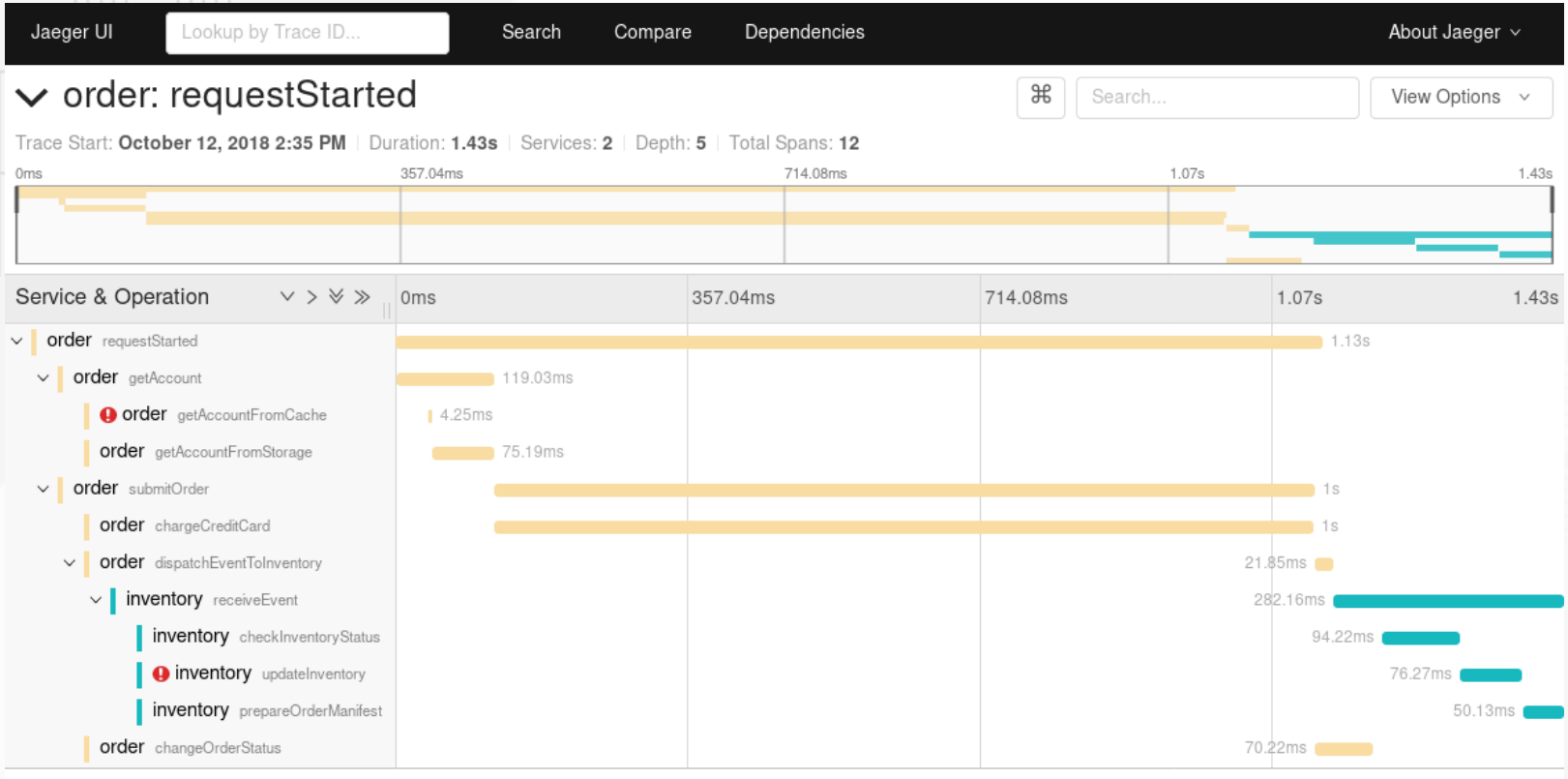
DISTRIBUTED TRACING

OUR CODE WITH DISTRIBUTED TRACING

AND CUSTOM TAG, WITH BUSINESS-SPECIFIC INFORMATION

```
try (Scope scope = tracer.buildSpan("submitOrder").startActive(true)) {
    scope.span().setTag("order-id", "c85b7644b6b5");
    chargeCreditCard();
    changeOrderStatus();
    dispatchEventToInventory();
}
```

DISTRIBUTED TRACING



OPENTRACING

SPECIFICATION

WHAT'S A TRACE, WHAT'S A SPAN, ...

INSTRUMENTATION API

OFFICIAL APIS FOR GO, JAVASCRIPT, JAVA, PYTHON, RUBY, PHP, OBJECTIVE-C, C++, C#

CAN BE USED WITH COMPATIBLE TRACERS, LIKE ZIPKIN, JAEGER, ...

OPENTRACING

SPAN

- OPERATION NAME
- START/FINISH TIMESTAMP
- TAGS, AS KEY-VALUE PAIRS
- LOGS
- REFERENCES (FOLLOWS-FROM, CHILD-OF)
- SPAN CONTEXT

OPENTRACING

SPAN CONTEXT

- BAGGAGE ITEMS
- IMPLEMENTATION-SPECIFIC STATE, LIKE:
  * TRACE ID
  * SPAN ID

THIS IS WHAT IS SENT ACROSS PROCESS BOUNDARIES (INJECTED/EXTRACTED)

OPENTRACING

TRACER

BUILDS SPANS CAPTURED BY THE INSTRUMENTATION

WRITES SPANS SOMEWHERE

PERFORMS INJECT/EXTRACT OPERATIONS

OPENTRACING

FRAMEWORK, STACK, PLATFORM

JAX-RS, JDBC, SERVLET, ...

SPRING BOOT, MICROPROFILE, BYTECODE MANIPULATION, ...

AND MORE

GITHUB.COM/OPENTRACING-CONTRIB

OPENTRACING

IS NOT A TRACING SOLUTION

NO BACKEND

NO WIRE SPEC

NO PROVISIONS FOR VENDOR INTEROPERABILITY

JAEGER

IT'S NOT OPENTRACING

CONCRETE OPENTRACING TRACER AND THE BACKEND COMPONENTS

Q&A