Jaeger FAQ

Updated January 22, 2018

JAEGER

CLOUD NATIVE COMPUTING FOUNDATION

TABLE OF CONTENTS

What is Jaeger?

Jaeger is a distributed tracing system built in the style of Google's Dapper. It consists of trace collection backend, web frontend and a set of OpenTracing compatible instrumentation libraries in multiple languages.

What is Distributed Tracing?

Distributed tracing is a technique for monitoring transactions in distributed systems. In the words of Ben Sigelman:

"With the transition to microservices, individual end-user requests often traverse dozens if not hundreds of system components; tools that consider each microservice in isolation simply do not tell clear stories anymore, and that's a massive operational problem.

Simply put, distributed tracing tells these stories about transactions in distributed systems; by following the transaction as it propagates from service to service, distributed tracing tools illuminate the relationship between user-visible behavior at the top of the stack and the complex mechanics of the distributed systems underneath."

Why is Jaeger needed in the market?

Distributed tracing techniques provide deep insights into the behavior of complex distributed systems. They are a tool of choice for many commercial APM vendors like Lightstep, Datadog and New Relic. Jaeger brings the power of commercial offerings as an open source project. By supporting CNCF's OpenTracing standard, Jaeger allows organizations to avoid vendor lock-in and choose a tracing system based on features and cost/benefit tradeoffs.

What problems does it solve?

Jaeger monitors distributed transactions occurring in complex, usually microservices-based, systems. It collects performance data, call graph shapes, contextual logs and exceptions, along with custom application events, while running in production continuously with minimal overhead. The collected tracing data can be used for root-cause analysis of outages, performance and latency optimization, analysis of service dependencies and many other purposes. Jaeger instrumentation provides a basis for generic distributed context propagation that can be used for cost accounting, chaos engineering, capacity planning and even partitioning regular metrics by the type of traffic.

Which companies are using it?

Companies around the world use Jaeger in production to monitor transactions within distributed systems, including Uber, Symantec, Red Hat, Base CRM, Massachusetts Open Cloud, Nets, Farmers Edge, Grafana Labs, Northwestern Mutual, Zenly, and more (see ADOPTERS.md).

What projects/technologies compete with Jaeger?

In open source, Jaeger's closest competitor is OpenZipkin. Jaeger also competes with commercial vendors like Lightstep, NewRelic and SolarWinds, along with SaaS offerings like Google's StackDriver and Amazon's X-Ray.

What sets Jaeger apart from traditional monitoring tools?

Traditional tools like metrics and logging help engineers monitor the behavior of individual services or components in isolation. Jaeger helps monitor transactions that span multiple components and tells a more comprehensive story about how the distributed system functions as a whole.

How was Jaeger inspired by Dapper and OpenZipkin? Which features of these technologies were built into & improved upon by Jaeger?

Moving into a microservices ecosystem brings a number of challenges, including the loss of visibility into the system and the complex interactions now occurring between services. The Uber Engineering team saw the reliability of the system as paramount, yet impossible without observability. While traditional monitoring tools still have their place, they often fail to provide visibility across services. This is where distributed tracing, namely Jaeger, thrives.

Functionally, Jaeger is very similar to Dapper (which is closed source) and OpenZipkin. For a period of time Jaeger actually used some components of OpenZipkin, but eventually they were re-implemented from scratch to work with Jaeger's richer data model.

Jaeger differs from and builds upon Zipkin in the following ways:

• All Jaeger instrumentation libraries are built to support the OpenTracing standard. While the Zipkin ecosystem has a few OpenTracing compatible libraries, many of them do not support it – instead requiring that applications use bespoke APIs and Zipkin-specific semantic annotations.

• Jaeger was built from the ground up with OpenTracing standard in mind, including the backend, the data models, and the UI. Zipkin's backend does not support every OpenTracing feature, specifically structured k-v span logs and multi-parent spans / DAGs (which are possible in OpenTracing via span references).

• All Jaeger libraries support context propagation via the "baggage" OpenTracing feature, as general purpose context propagation is extremely useful in microservices-based systems. In addition to tracing, it can support many other applications, including resource attribution or chargebacks, capacity planning, chaos engineering and security.

How do OpenCensus and Jaeger differ?

OpenCensus is a collection of instrumentation libraries in different languages that combine tracing and metrics APIs. Where OpenTracing only defines pure APIs that allow a lot of freedom to concrete implementations, OpenCensus libraries are specific, opinionated implementations. They do allow some flexibility in plugging in different exporters for different monitoring backends, provided that those backends conceptually match the internal data model of OpenCensus. For example, OpenCensus includes reporters for Zipkin backend and can be used with Jaeger backend, as Jaeger is backward compatible with Zipkin.

Where can I find more information about Jaeger?

Visit http://jaeger.readthedocs.io or http://jaegertracing.io for more information on the project.

For educational, technical and case study presentations about Jaeger, check out:

- KubeCon + CloudNativeCon North America 2017 Session: Would You Like Some Tracing With Your Monitoring? (Slides)
- The original blog post introducing Jaeger: “Evolving Distributed Tracing at Uber”
- JaegerTracing blog on Medium.com
- @JaegerTracing on Twitter
- Jaeger documentation site

What does it mean to be a CNCF Project?

As a CNCF hosted project, Jaeger is part of a neutral community aligned with technical interests to help companies move to cloud native deployment models and help developers deliver on the promise of microservices and cloud native applications at scale. As Jaeger grows, CNCF is helping build its community, marketing and documentation efforts. For more, read “CNCF Hosts Jaeger.”

Does Jaeger compete with any CNCF projects? If so, which ones?

Jaeger does not currently compete with any CNCF projects. It provides an implementation of OpenTracing APIs in several languages, and complements Prometheus and Fluentd as a monitoring tool.

Why does CNCF have competing projects?

The cloud native ecosystem is still nascent and quickly evolving. CNCF recognizes that many promising technologies are emerging, some better suited for certain workloads, use cases or environments. CNCF’s goal is help many rising technologies mature. In this respect, we expect to have overlapping projects in some cases.