OpenTracing

textkernel

Machine Intelligence for People and Jobs

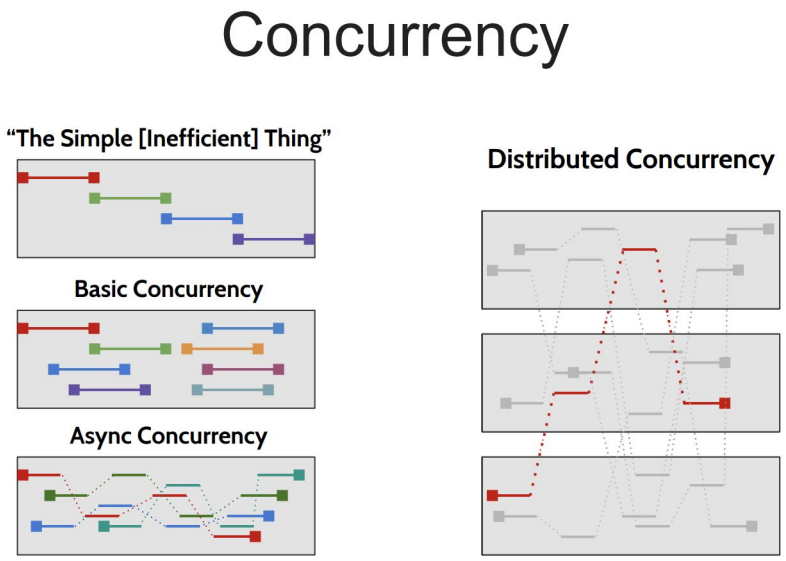
Amsterdam Gomeetup

About Me

Mehmet Koray Sariteke

SW Developer @Textkernel

What is Distributed Tracing



- Distributed tracing (request tracing) is a method used to profile & monitor applications.
- It helps pinpoint where failures occur and what causes poor performence

Monolith

Microservices

Illustration by Lev Polyakov, http://levpolyakov.com

What is OpenTracing

- OpenTracing is an API specifications. Distributed tracing requires that software developers add instrumentation to the code of an application, or to the frameworks used in the application.
- **OpenTracing is not a standard?** The Cloud Native Computing Foundation (CNCF) is not an official standards body. The OpenTracing API project is working towards creating more standardized APIs and instrumentation for distributed tracing.

Good News!! - OpenTelemetry

OpenTelemetry makes robust, portable telemetry a built-in feature of cloud-native software, and is the next major version of both OpenTracing and OpenCensus.

https://github.com/open-telemetry/opentelemetry-specification

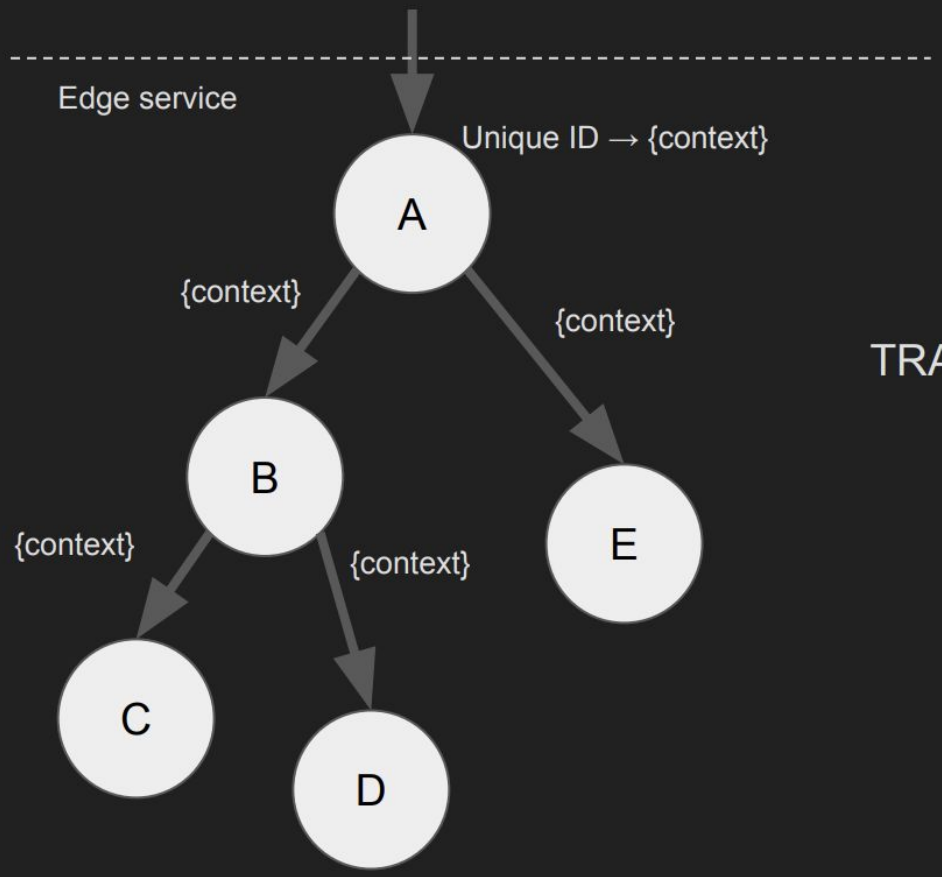
https://opentelemetry.io/

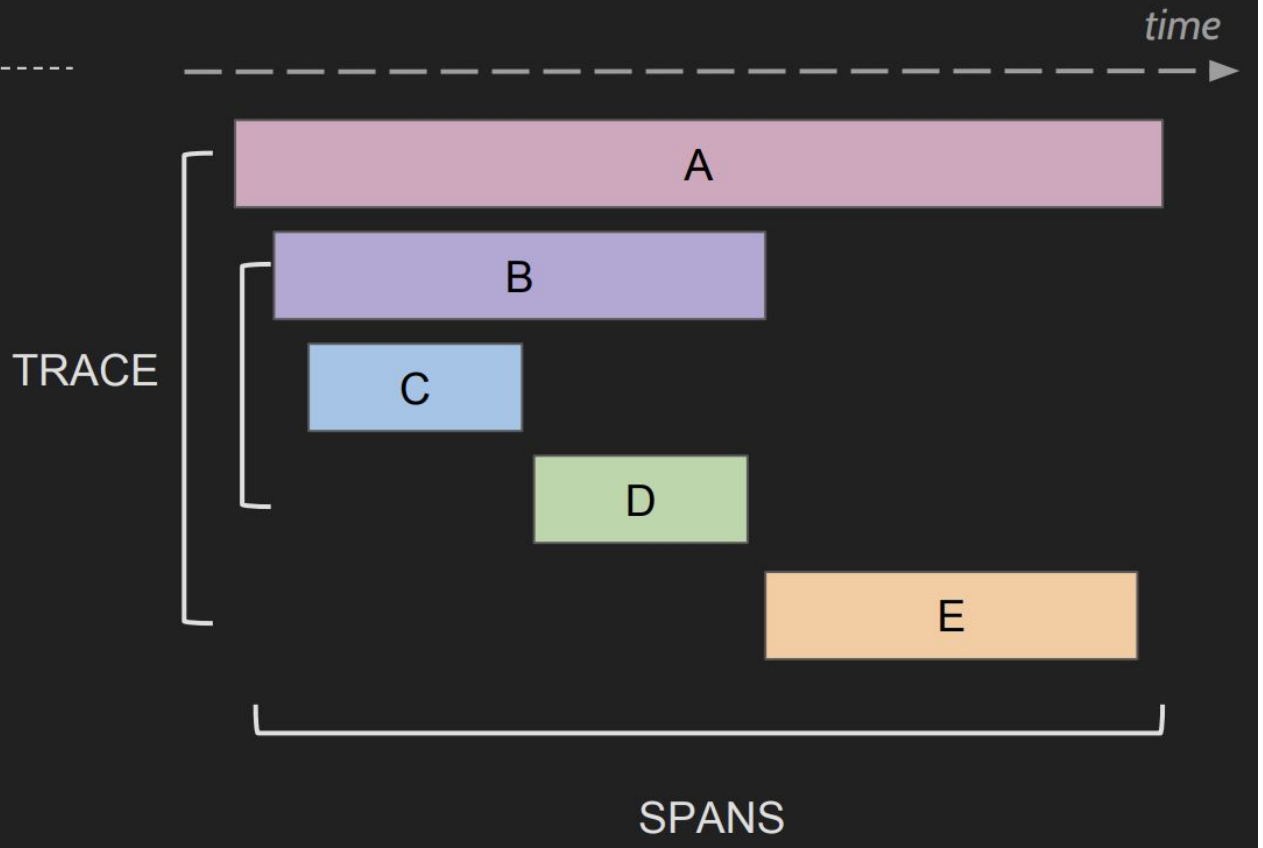
https://medium.com/opentelemetry

https://medium.com/jaegertracing/jaeger-and-opentelemetry-1846f701d9f2

The greatest promise of OpenTelemetry is a single standard for observability instead of two competing standards.

Distributed Context Propagation and Tracing





Run Jaeger All for testing purpose

```
docker run -d --name local-jaeger \
-e COLLECTOR_ZIPKIN_HTTP_PORT=9411 \
-p5775:5775/udp \
-p6831:6831/udp \
-p6832:6832/udp \
-p5778:5778 \
-p16686:16686 \
-p14268:14268 \
-p9411:9411 \
jaegertracing/all-in-one:latest
```

Extract Span

```
tracer := opentracing.GlobalTracer()
wireCtx, err := tracer.Extract(
    opentracing.HTTPHeaders,
    opentracing.HTTPHeadersCarrier(r.Header))

var serverSpan opentracing.Span
if err == nil {
    serverSpan = opentracing.StartSpan(r.URL.Path,
        ext.RPCServerOption(wireCtx))
} else {
    serverSpan = opentracing.StartSpan(r.URL.Path)
}
```

Inject into request Header

```
tracer.Inject(serverSpan.Context(),
    opentracing.HTTPHeaders,
    opentracing.HTTPHeadersCarrier(r.Header))

req, _ := httputil.DumpRequest(r, true)
log.Info(string(req))

h.ServeHTTP(w, r)
```

ChildOf span

```
        tracer := opentracing.GlobalTracer()
        oCtx, err := tracer.Extract(opentracing.HTTPHeaders,
opentracing.HTTPHeadersCarrier(r.Header))

        var span opentracing.Span
        if err == nil {
            span = tracer.StartSpan("upsert-birthday", opentracing.ChildOf(oCtx))
        } else {
            span = tracer.StartSpan("upsert-birthday")
        }
        defer span.Finish()
        spanCtx := opentracing.ContextWithSpan(r.Context(), span)
```

Span Tag

```
span.SetTag("http.method", r.Method)
```

- The tag key, which must be a string
- The tag value, which must be either a string, a boolean value, or a numeric type

DEMO…

References

- https://github.com/ksaritek/opentracing-sample
- https://www.youtube.com/watch?v=fjYAU3jayVo
- https://www.amazon.com/Mastering-Distributed-Tracing-performance-microservices-ebook/dp/B07MBNGF7Q
- https://opentracing.io

Questions?

?