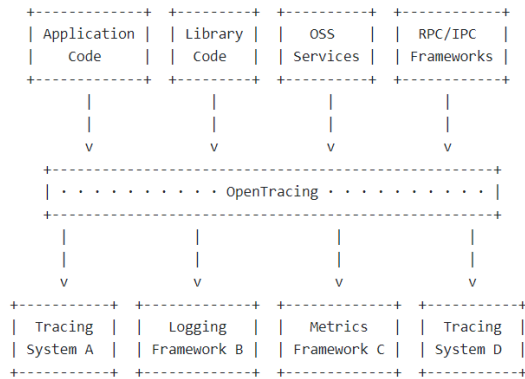


OpenTracing

Opentracing

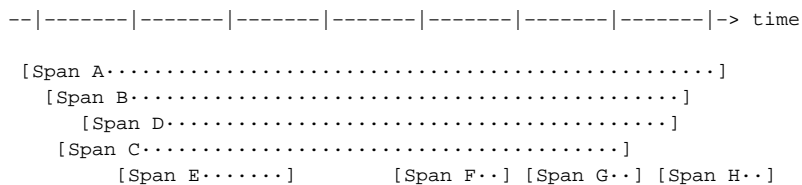
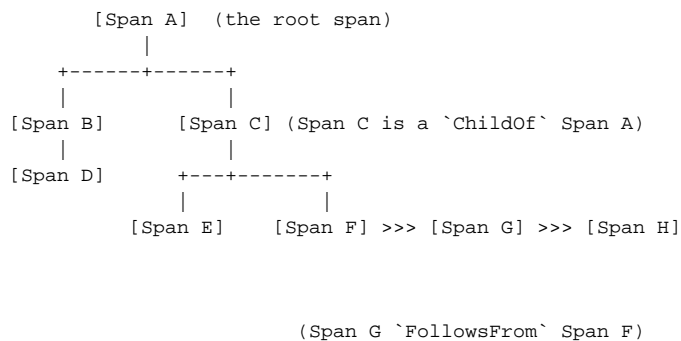


OpentracingCNCFOpentracingAPIGoOpentracinginterfaceOpentracingOpentracingOpentracing<https://github.com/opentracing/opentracing-go>

Opentracing

Trace

Tracetracespan8spantrace:



<https://github.com/opentracing/specification/blob/master/specification.md#the-opentracing-data-model>

Span

Spanspanhttpspan:

- operation name
-
- K/VTags
- K/VLogs

Content Menu

- **Opentracing**
 - **Trace**
 - **Span**
 - **Tags**
 - **Logs**
 - **SpanContext**
 - **References**
- - **Jaeger**
 - **Quick Start**
 -

- SpanContext
- ReferencesspanspanSpanContext

Tags


TagsK/Vhttp.method="GET",http.status_code=200keyvalue spantag SpanContextspan

```
span.SetTag("http.method","GET")
span.SetTag("http.status_code", 200)
```

Logs

LogtagsK/Vtagslogslogslogslogskeyvalue

```
span.LogFields(
    log.String("event", "soft error"),
    log.String("type", "cache timeout"),
    log.Int("waited.millis", 1500),
)
```

 OpentracingTagsLogshttps://github.com/opentracing/specification/blob/master/semantic_conventions.md

SpanContext

SpanContext


- spanspan_id, trace_id
- Baggage ItemsK/VBaggage ItemtagsK/Vtags
 - keyvalue
 - Baggage itemsspanSpanContextspanbaggage items - spanK,VCPU

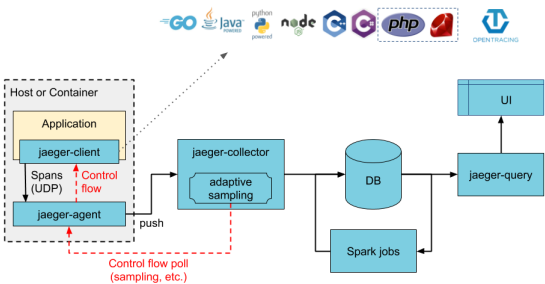
References

Opentracing:ChildOfFollowFrom

- ChildOf: spansspanspanspanspanChildOfRPCspanspanspanspanspanspanChildOf
- FollowFromspanspanspanspanspanspanFollowFromFollowFromconsumer spanproducer span

OpentracingJaeger

 [Opentracing Go Tutorial](#)



Jaeger

[Jaeger](#)ly-gr\ UberOpentracingCNCFJaeger

Quick Start

Jaegerall-in-one

```
docker run -d --name jaeger \  
-e COLLECTOR_ZIPKIN_HTTP_PORT=9411 \  
-p 5775:5775/udp \  
-p 6831:6831/udp \  
-p 6832:6832/udp \  
-p 5778:5778 \  
-p 16686:16686 \  
-p 14268:14268 \  
-p 9411:9411 \  
jaegertracing/all-in-one:1.14
```

<http://localhost:16686> Jaeger UI

library:

```
go get github.com/jaegertracing/jaeger-client-go
```

Jaeger tracer:

```
import (  
    "context"  
    "errors"  
    "fmt"  
    "io"  
    "time"  
  
    "github.com/opentracing/opentracing-go"  
    "github.com/opentracing/opentracing-go/log"  
    "github.com/uber/jaeger-client-go"  
    jaegerCfg "github.com/uber/jaeger-client-go/config"  
)  
  
// initJaeger jaeger tracertracer  
func initJaeger(service string) io.Closer {  
    cfg := jaegerCfg.Configuration{  
        // lspan  
        Sampler: &jaegerCfg.SamplerConfig{  
            Type: jaeger.SamplerTypeConst,  
            Param: 1,  
        },  
        Reporter: &jaegerCfg.ReporterConfig{  
            LogSpans: true,  
            // spanjaeger-collector  
            CollectorEndpoint: "http://localhost:14268/api  
/traces",  
        },  
    }  
    closer, err := cfg.InitGlobalTracer(service, jaegerCfg.Logger  
(jaeger.StdLogger))  
    if err != nil {  
        panic(fmt.Sprintf("ERROR: cannot init Jaeger: %v\n", err))  
    }  
    return closer  
}
```

tracerroot span

```
func main() {
    closer := initJaeger("in-process")
    defer closer.Close()
    // jaeger tracer
    tracer := opentracing.GlobalTracer()
    // root span
    span := tracer.StartSpan("in-process-service")
    // mainspan
    defer span.Finish()
    // spanFoo
    ctx := opentracing.ContextWithSpan(context.Background(), span)
    Foo(ctx)
}
```

root spanspancontextFooFoo

```
func Foo(ctx context.Context) {
    // span, spanoperation_nameFoo
    span, ctx := opentracing.StartSpanFromContext(ctx, "Foo")
    defer span.Finish()
    // contextBar
    Bar(ctx)
    //
    time.Sleep(1 * time.Second)
}

func Bar(ctx context.Context) {
    // spanspanoperation_nameBar
    span, ctx := opentracing.StartSpanFromContext(ctx, "Bar")
    defer span.Finish()

    //
    time.Sleep(2 * time.Second)

    // Bar
    err := errors.New("something wrong")
    span.LogFields(
        log.String("event", "error"),
        log.String("message", err.Error()),
    )
    span.SetTag("error", true)
}
```

FooBarBarspan.LogFieldsspan.SetTag context.Contextopentracing.ContextWithSpancontextopentracing.StartSpanFromContextspan

Jaeger UI: <http://localhost:16686/search>

spantaglog

- [Opentracing Go Tutorial](#)
- [Opentracing](#)
- [Opentracingv1.1](#)

