I see what you did there!
Tracing in Go

Maximilian Bischoff,
Johannes M. Scheuermann
Maximilian Bischoff
Cloud Platform Engineer
@bisch_off_max

Johannes M. Scheuermann
EDV Experte Profi
@johscheuer
What about you?
Why should I use tracing?
Why should I use distributed tracing?

- Instrumentation doesn’t show causality
  - or it’s pretty hard (logs)
- Main feature is correlation between events

- “I want to know why call X didn’t work”
- “Why is Y so slow?”
- “How do we optimize the performance”
Tracing basics
Tracing Basics

A trace is a data/execution path through the system
Tracing Basics

Anatomy of a Span

<table>
<thead>
<tr>
<th>Tag</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>http.method</td>
<td>&quot;GET&quot;</td>
</tr>
<tr>
<td>http.status_code</td>
<td>200</td>
</tr>
<tr>
<td>http.url</td>
<td>&quot;/toastoftheday&quot;</td>
</tr>
<tr>
<td>internal.span.format</td>
<td>&quot;jaeger&quot;</td>
</tr>
<tr>
<td>sampler.param</td>
<td>true</td>
</tr>
<tr>
<td>sampler.type</td>
<td>&quot;const&quot;</td>
</tr>
<tr>
<td>span.kind</td>
<td>&quot;client&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Process</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>client-uuid</td>
<td>&quot;2a48d7a959457f63&quot;</td>
</tr>
<tr>
<td>hostname</td>
<td>&quot;502ae8af96da&quot;</td>
</tr>
<tr>
<td>ip</td>
<td>&quot;172.21.0.4&quot;</td>
</tr>
<tr>
<td>jaeger.version</td>
<td>&quot;Go-2.22.0&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Logs (1)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.44ms: Toasts</td>
<td>([Hawaii Monday] (Peperoni Tuesday) (Cheese Wednesday) (Ham Thursday) (Caprese Friday) (Avocado Saturday) (Honey Sunday))</td>
</tr>
</tbody>
</table>

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

Service B

Database

Trace: xyz

traceld: xyz
spanld: 123
parentld: nil

Service A
add_usr
id: 123
Service A

add_usr

Service B

post_users

Database

POST /users

traceld: xyz

parentId: 123

traceId: xyz

spanId: 123

parentId: nil

traceId: xyz

spanId: 456

parentId: 123

Trace: xyz

Service A

add_usr

id: 123

Service B

post_users

id: 456

https://www.w3.org/TR/trace-context-1/#trace-id
Trace: xyz

Service A
- add_usr
  - traceId: xyz
  - parentId: 123

Service B
- post_users
  - traceId: xyz
  - parentId: 456

Database
- insert_user
  - traceId: xyz
  - parentId: 456

POST /users
  - traceId: xyz
  - parentId: 123

Tracing Backend

POST /users
  - traceId: xyz
  - parentId: 123

Tracing

POST /users
  - traceId: xyz
  - parentId: nil

Database
  - traceId: xyz
  - spanId: 456
  - parentId: 123

Tracing Backend

Database

POST /users
  - traceId: xyz
  - spanId: 789
  - parentId: 456

Tracing Backend

Database

Tracing

POST /users
  - traceId: xyz
  - spanId: 456
  - parentId: 456

Tracing Backend

Database

Tracing

POST /users
  - traceId: xyz
  - spanId: 123
  - parentId: 456

Tracing Backend

Database

Tracing

POST /users
  - traceId: xyz
  - spanId: nil
  - parentId: 456

Tracing Backend

Database

Tracing

POST /users
  - traceId: xyz
  - spanId: 789
  - parentId: 456

Tracing Backend

Database

Tracing

POST /users
  - traceId: xyz
  - spanId: 123
  - parentId: 456
Tracing in Go
Demo Architecture

/toastoftheday -> frontend

/toasts -> backend
Recap
Recap

- Brings a huge value
  - Very useful in distributed systems
- Relativ simple to implement
- Can produce quite a lot of data
- Definitely worth a look
- OpenTracing offers a bunch of libraries
Links

● [https://github.com/inovex/godays-demo](https://github.com/inovex/godays-demo)
● [https://opentracing.io](https://opentracing.io)
● [https://opentracing.io/guides/golang/quick-start](https://opentracing.io/guides/golang/quick-start)
● [https://opentelemetry.io](https://opentelemetry.io)
Thank You

Maximilian Bischoff
IT Engineering & Operations
inovex GmbH
Ludwig-Erhard-Allee 6
76131 Karlsruhe
maximilian.bischoff@inovex.de

Johannes Scheuermann
IT Engineering & Operations
inovex GmbH
Ludwig-Erhard-Allee 6
76131 Karlsruhe
johannes.scheuermann@inovex.de