Towards Cross-Layer Telemetry

Justin Iurman, Frank Brockners, Benoit Donnet

ANRW’21, July 26-30 2021 (virtual)
Microservices

**MONOLITHIC**

- UI
- Business Logic
- Data Access Layer

VS.

**MICROSERVICES**

- UI
- Microservice
- Microservice
- Microservice
Still confident to debug this one?
Application Performance Management

- Distributed tracing tools (e.g. Jaeger)
- Very useful for (spaghetti) microservices
Application Performance Management

• BUT... what if my DB lookup is slow?
  – Is it the app?
  – Is it the server, or the DB?
  – Is it a network issue?
  – Is it Chuck Norris’ fault?

→ We don’t know why or where the problem exactly is, we only know it is slow.
Cross-Layer Telemetry

- **Solution**: include L3-L4 in tracing tools visibility
- (1) Find a way to correlate APM (e.g. Jaeger) traces with network traffic
  - use IOAM to carry both trace and span IDs in packets
- (2) Find *when* and *how* these IDs should be injected
  - (when?) at socket creation vs when sending data
  - (how?) send* API modification vs new syscall vs netlink call
Cross-Layer Telemetry

Service & Operation

Login API span_login

Tags: internal.span.format = "proto"; sampler.param = true; sampler.type = const

Process: hostname: c297503535d5; ip: 192.168.224.4; jaeger.version: Python 4.3.0

Login API ioam_span

Tags:
- internal.span.format = "proto"
- ioam_namespace123_node1 = "Hop Limit=64; Node_Id=1; Ingress_Id=65535; Egress_Id=11; Egress_Queue_Depth=0;"
- ioam_namespace123_node2 = "Hop Limit=63; Node_Id=2; Ingress_Id=21; Egress_Id=22; Egress_Queue_Depth=97;"
- ioam_namespace123_node3 = "Hop Limit=62; Node_Id=3; Ingress_Id=31; Egress_Id=65535; Egress_Queue_Depth=0;"
- sampler.param = true

(Entry Point) (Router) (Server)
Conclusion

• Hot topic in the industry
• CLT solves challenges in the (microservice) tracing world
• Next steps to improve it
• Github: https://github.com/iurmanj/cross-layer-telemetry (+ demo video)
Thank you!

justin.iurman@uliege.be