Distributed Tracing OpenTracing

“distributed applications under control”

Pavol Loffay

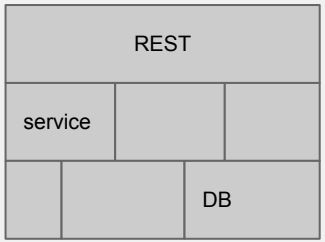
5.4.2017

Agenda

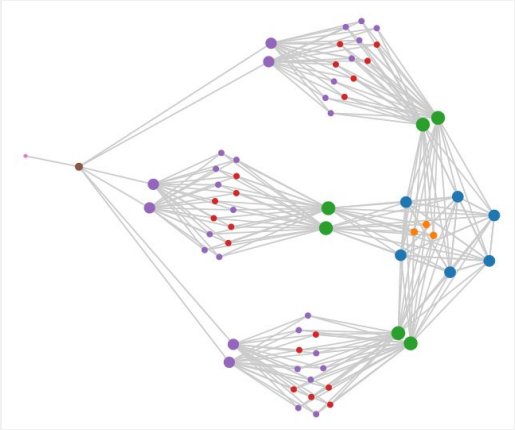
- Why care about distributed tracing?
- Concepts and terminology
- Demo + live coding… best practices
- How to start with OpenTracing

Why Tracing?

Monolithic vs. Microservices



Distributed systems break!



How do you monitor? Does your monitoring tell story?

1 story vs. N storytellers

... cont.

- X services
- Y frameworks
- Z languages

....instrumentation is the hardest part of a tracing deployment

OpenTracing

- It is only API!
  - **explicit** & **decoupled** & **consistent**
- no data format!
- no wire format!

- consistently models and describes the behavior of distributed systems

Why [Open]Tracing?

- **Logging** - Easy to output to any logging tool
- **Metrics/Alerting** - Measure based on tags, span timing, log data
- **Context Propagation** - Use baggage to carry request and user ID's, etc.

- **Critical Path Analysis** - Drill down into request latency in very high fidelity
- **System Topology Analysis** - Identify bottlenecks due to shared resources

OpenTracing Implementations

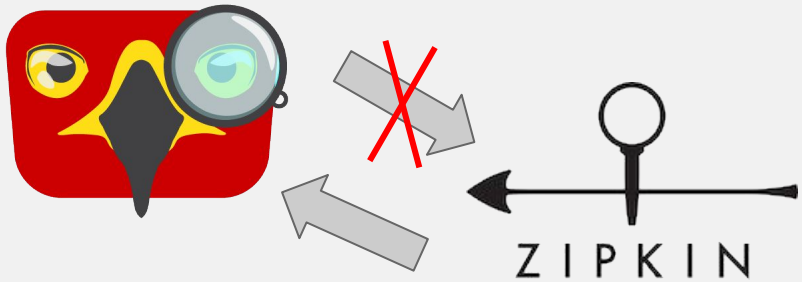
Hawkular APM - java, js, more coming soon...

Zipkin - java, go

Jaeger - java, js, go, python

appdash - go, python, ruby

.. additional smaller projects

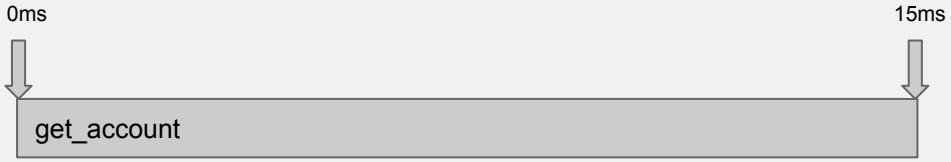


OpenTracing

- started in 2015
- API in 8 languages: Go, Java, JS, Python, Ruby, C++, C#, Objective-C
- version 1.0 announced in August 2016, 2.0 this year!
- Cloud Native Computing Foundation

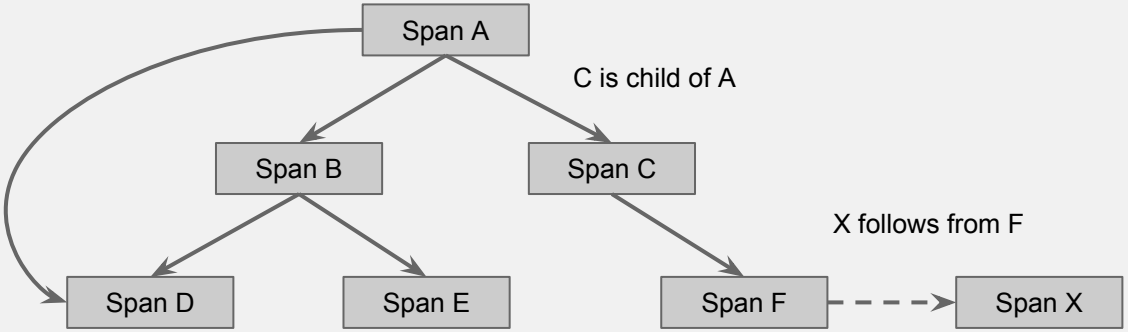
Span

- represents a "unit of work"
- start / end
- tags
- logs (timed event)

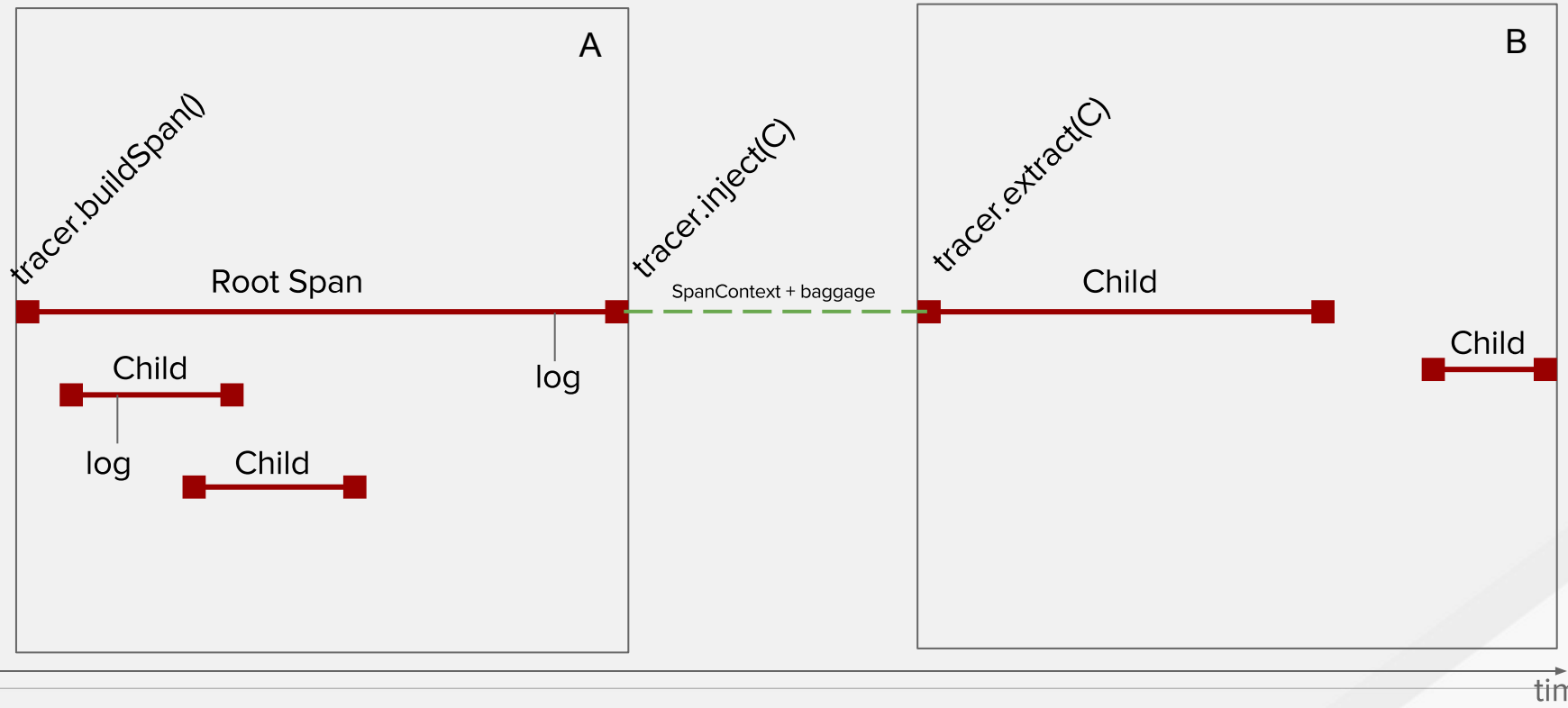


Trace

- list of spans
- tree-like structure (DAG)



OpenTracing API



OpenTracing API

Tracer

- `Span = tracer.buildSpan(name)`
- `SpanContext = tracer.extract(carrier)`
- `tracer.inject(SpanContext, carrier)`

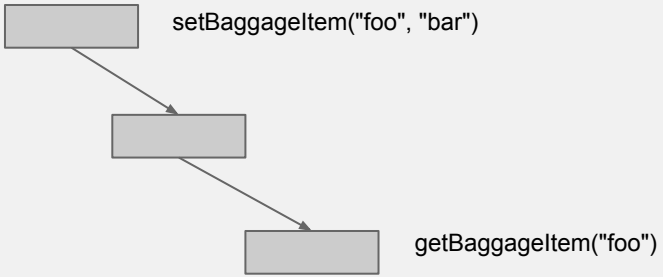
OpenTracing API

SpanContext

- `baggageItems()`

Span

- `setTag(K, V)`
- `log(timestamp, map)`
- `finish()`



Demo

Sampling

- implementation specific!

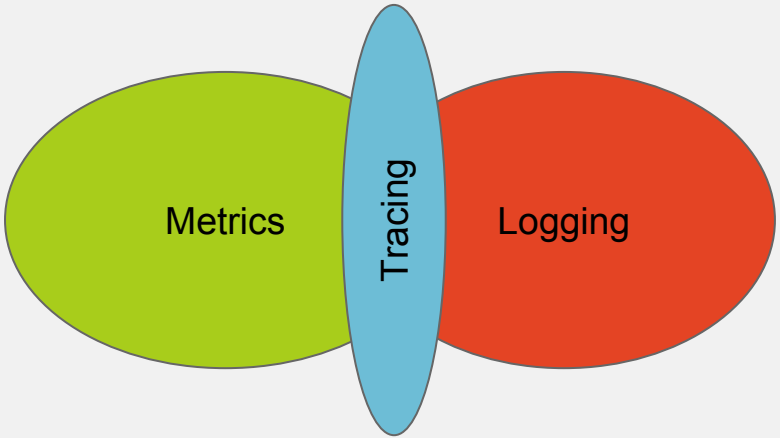
- adaptive sampling
- post-trace sampling
- based on data/tags (errors?)

Instrumentations

- explicit
- framework integrations
- non intrusive e.g. -javaagent

Conclusion

- instrumentation is hard
- start spans at the right places



Getting involved

Chat: gitter.im/opentracing/public

Github: github.com/opentracing

Web: opentracing.io

Chat: freenode #hawkular

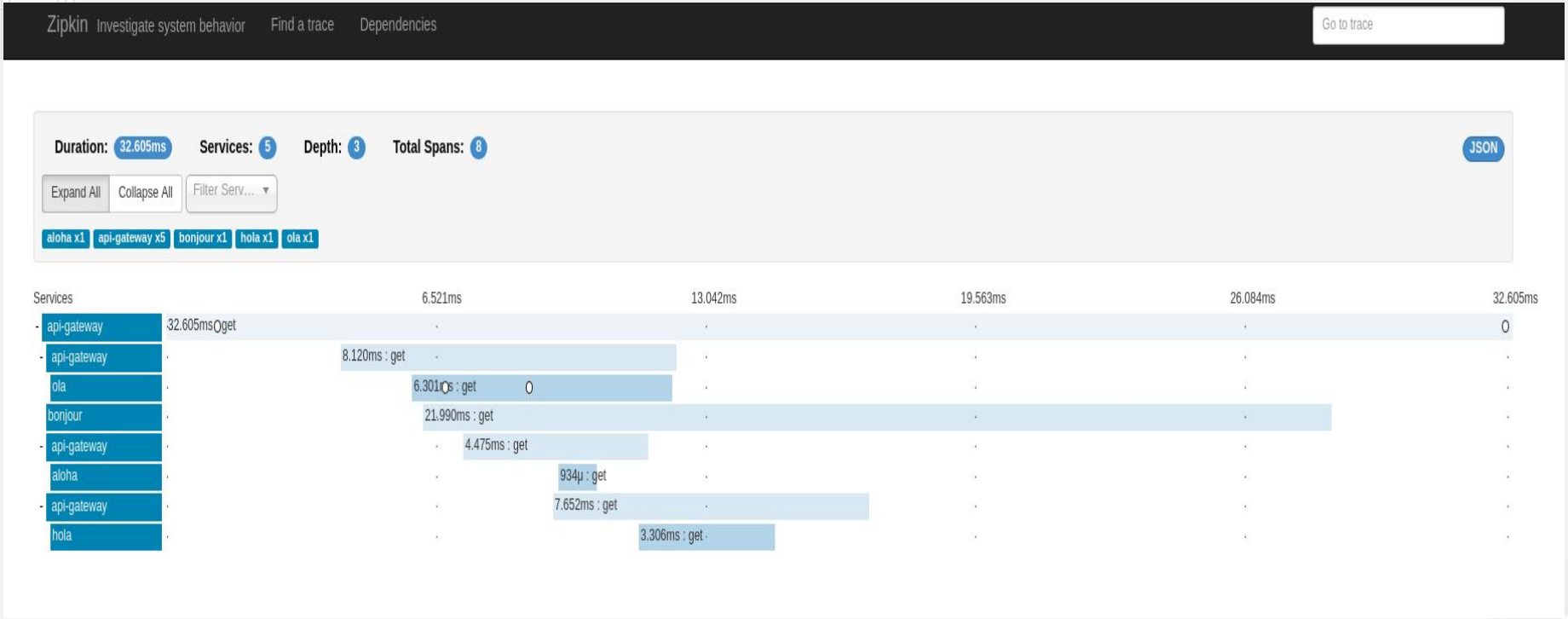
Github: github.com/hawkular

Web: hawkular.org

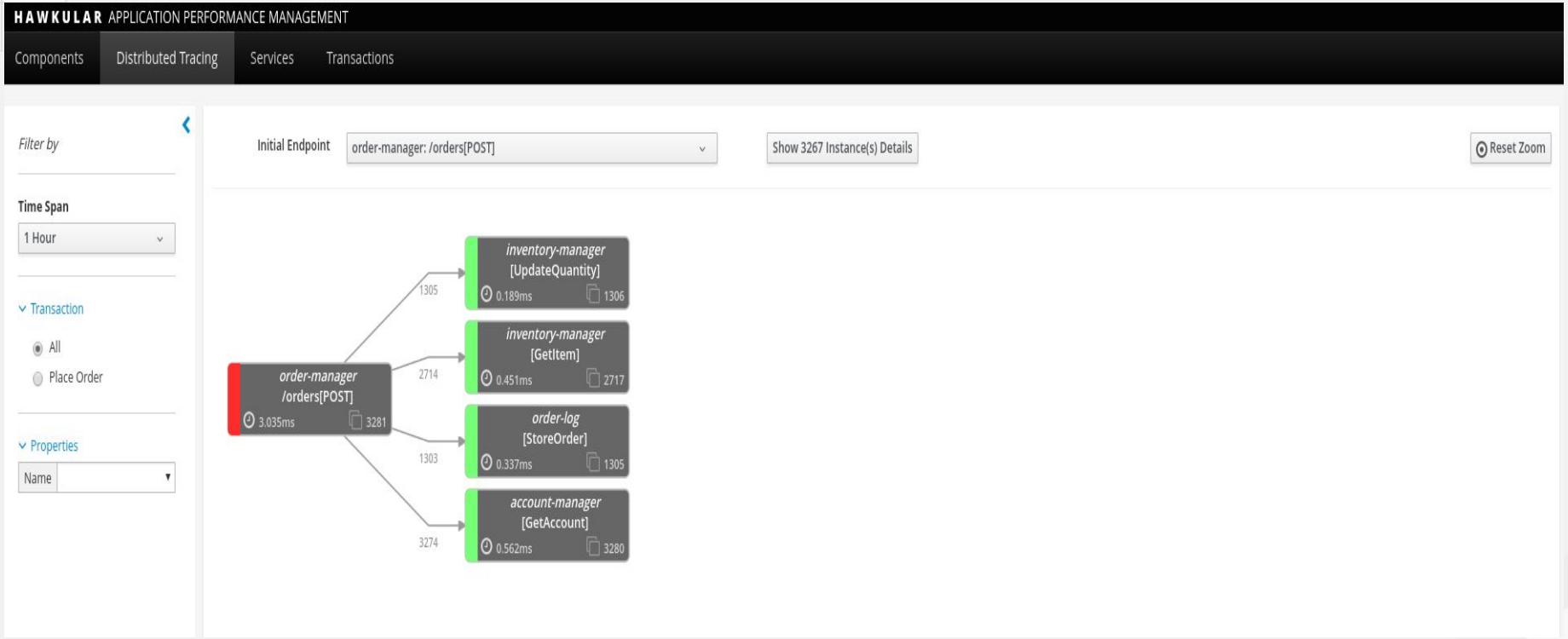
Thanks / Q&A

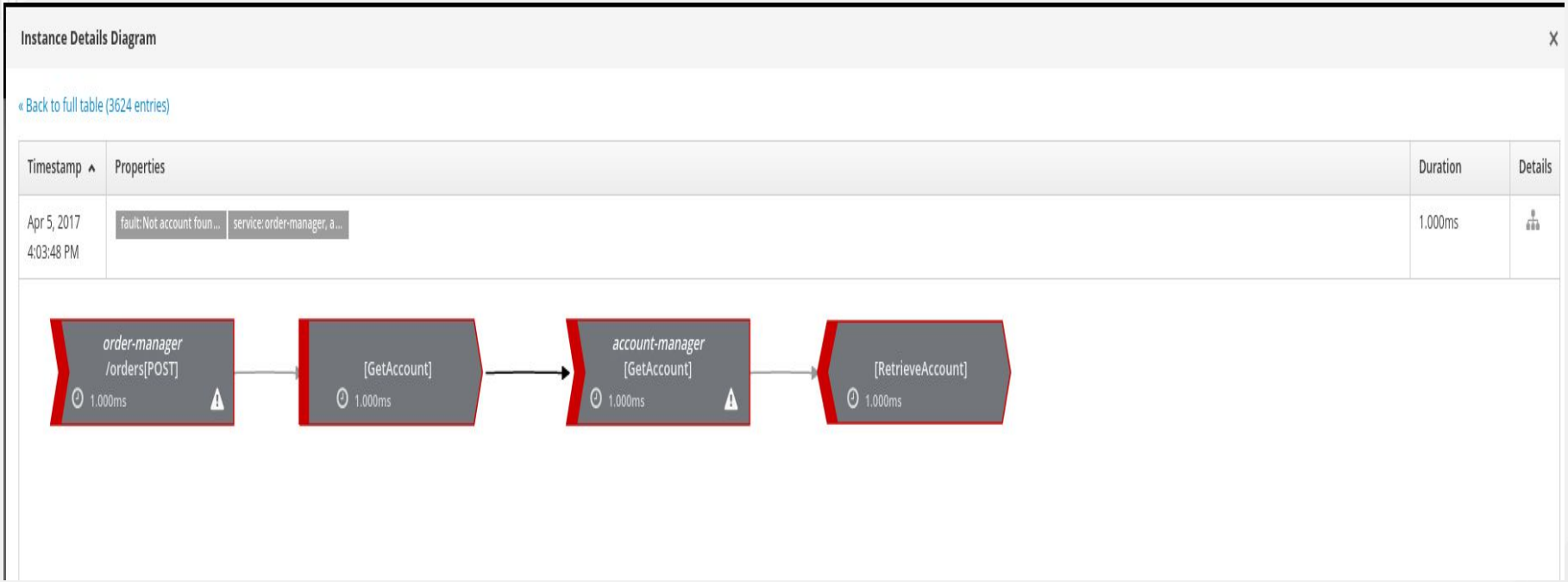
Pavol Loffay

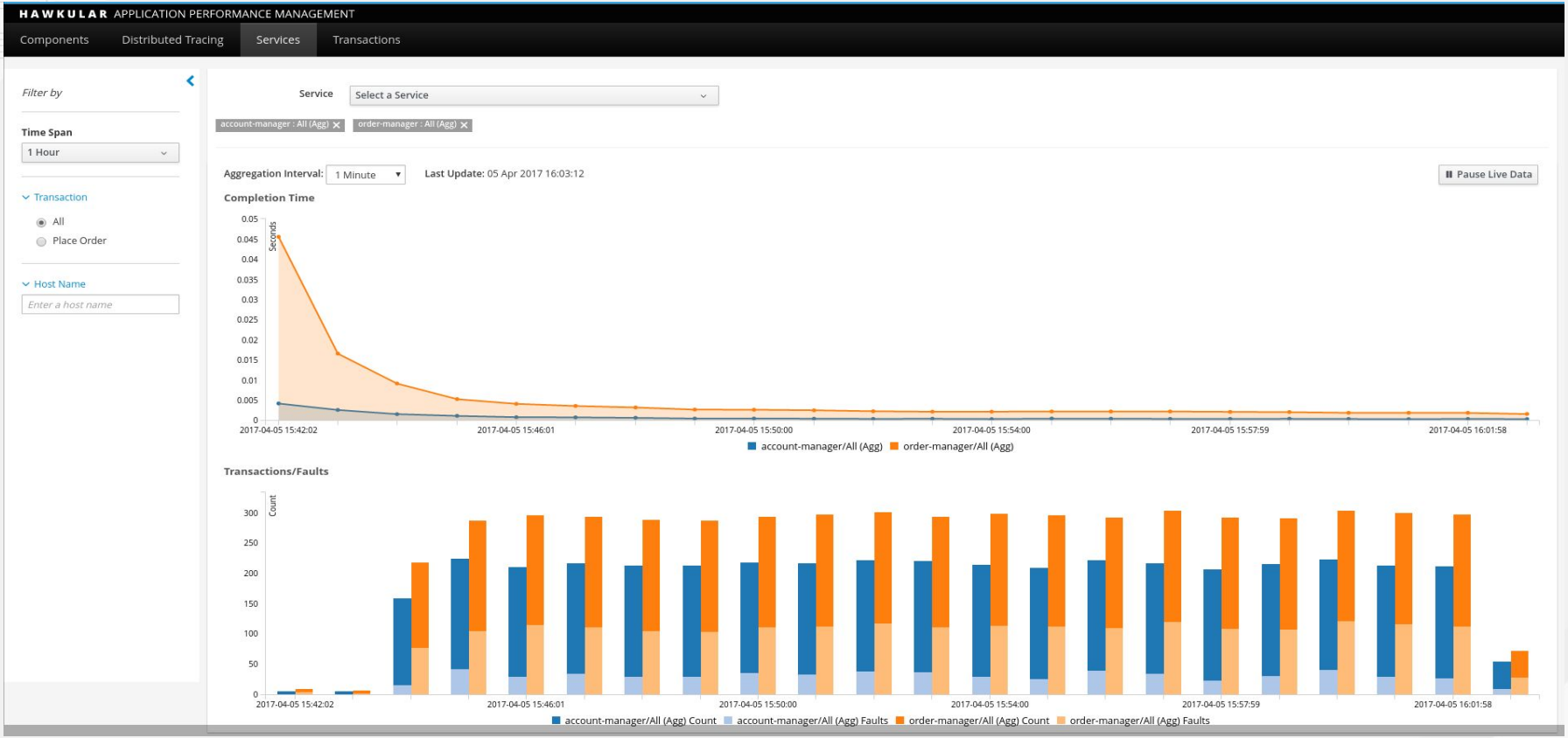
ploffay@redhat.com / @pavolloffay


Zipkin
Investigate system behavior
Find a trace
Dependencies
Go to trace
Duration: 32.605ms
Services: 5
Depth: 3
Total Spans: 8
JSON
Expand All
Collapse All
Filter Serv...
aloha x1
api-gateway x5
bonjour x1
hola x1
ola x1
Services
6.521ms
13.042ms
19.563ms
26.084ms
32.605ms
api-gateway
32.605ms : get
api-gateway
8.120ms : get
ola
6.301ms : get
bonjour
21.990ms : get
api-gateway
4.475ms : get
aloha
934µ : get
api-gateway
7.652ms : get
hola
3.306ms : get


Zipkin
Investigate system behavior
Find a trace
Dependencies
Go to trace
Duration: 50.067s
Services: 2
Depth: 4
Total Spans: 8
JSON
Expand All
Collapse All
Filt...
hola x6
ola x2
Services
10.013s
20.027s
30.040s
40.054s
50.067s
ola
909011ms : get
ola
1.013s : get
hola
1.010s : get
hola
10.012s : get
hola
10.012s : get
hola
10.012s : get
hola
10.005s : get
hola
10.011s : get


HAWKULAR APPLICATION PERFORMANCE MANAGEMENT
Components
Distributed Tracing
Services
Transactions
Filter by
Time Span
1 Hour
Transaction
All
Place Order
Properties
Name
Initial Endpoint
order-manager: /orders[POST]
Show 3267 Instance(s) Details
Reset Zoom
order-manager
/orders[POST]
3.035ms
3281
1305
inventory-manager
[UpdateQuantity]
0.189ms
1306
2714
inventory-manager
[GetItem]
0.451ms
2717
1303
order-log
[StoreOrder]
0.337ms
1305
3274
account-manager
[GetAccount]
0.562ms
3280


Instance Details Diagram
« Back to full table (3624 entries)
Timestamp
Properties
Duration
Details
Apr 5, 2017
4:03:48 PM
fault:Not account foun...
service:order-manager, a...
1.000ms
order-manager
/orders[POST]
1.000ms
[GetAccount]
1.000ms
account-manager
[GetAccount]
1.000ms
[RetrieveAccount]
1.000ms


HAWKULAR APPLICATION PERFORMANCE MANAGEMENT
Components
Distributed Tracing
Services
Transactions
Filter by
Time Span
1 Hour
Transaction
All
Place Order
Host Name
Enter a host name
Service
Select a Service
account-manager : All (Agg)
order-manager : All (Agg)
Aggregation Interval:
1 Minute
Last Update: 05 Apr 2017 16:03:12
Pause Live Data
Completion Time
Seconds
0.05
0.045
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
2017-04-05 15:42:02
2017-04-05 15:46:01
2017-04-05 15:50:00
2017-04-05 15:54:00
2017-04-05 15:57:59
2017-04-05 16:01:58
account-manager/All (Agg)
order-manager/All (Agg)
Transactions/Faults
Count
300
250
200
150
100
50
0
2017-04-05 15:42:02
2017-04-05 15:46:01
2017-04-05 15:50:00
2017-04-05 15:54:00
2017-04-05 15:57:59
2017-04-05 16:01:58
account-manager/All (Agg) Count
account-manager/All (Agg) Faults
order-manager/All (Agg) Count
order-manager/All (Agg) Faults