Beyond Distributed Tracing

SREcon22 Americas
March 15 2022
Speaker: Kyusoon Lee (qsoonlee@google.com)
CUI Attribution team @ Google

Kyusoon
qsoonlee@google.com

Jan-Jan
janjan@google.com
Featuring works by
2020-11-11

“YouTube is down”
Incident log

T₀  First automated alert (SLO fast burn)
T₂  Another YouTube alert
T₄  Escalation to YT oncall “YouTube is down”
T₅  Escalation to YT tech lead, “YouTube is down”
T₈  Escalation to YT leadership, “large traffic increases, lots of things down”
T₉  More alerts
T₉  Suspects an ongoing push; proceeds to revert
T₁₀ Unusual spike in traffic reported
T₁₀ “Frontend is seeing errors from service X”
T₁₁ “What’s happening is service X is down”
T₁₂ “What is service X in this context”
T₁₂ More alerts
T₁₃ “Cluster y seems to be crashing a lot”
T₁₃ “going to page service X oncall”
T₁₄ Yet more alerts
T₁₄ First external comms: “working to fix it”

...
“YouTube doesn’t work”
Pick a product

**youtube_main_app**

---

Pick a CUI

**WATCH**

---

Pick a metric

**Error percentage (ser...**

---

**Query**
Server metrics

- All requests that go through the server
- One-hop only
Distributed tracing

- Per-request
- Multi-hop (end-to-end)

From “Dapper, a Large-Scale Distributed Systems Tracing Infrastructure”
https://research.google/pubs/pub36356/

See also https://cloud.google.com/trace/docs/reference
Server metrics
- All requests
- One-hop

Dist. tracing
- Per-request
- Multi-hop
Server metrics
- All requests
- One-hop

Dist. tracing
- A set of requests
- Multi-hop
- Per-request
  - Multi-hop
CUI (Critical User Interaction) attribution

CUI: a frequently used or highly important end-user interaction. Also used to refer the metadata.

**Format:** "<product>/<interaction>"

**Example:**
youtube/WATCH

**Baggage:** key-value pairs propagated with a distributed request

**Reference:**
https://www.w3.org/TR/baggage/

**Example:**
baggage: productCui=youtube/WATCH
CUI attribution

- Per-CUI
- Multi-hop (end-to-end)
Night and day difference
Google internal
Root-causing
an issue caused by
a service multiple hops away
Potential root-causing workflow?

Check incoming requests on the server → Check outgoing requests to the backends → Suspicious backend?

- Yes: The server is likely to be a culprit
- No: Go to the suspicious backend
Without CUI attribution
Without CUI attribution

With CUI attribution
Errors returning from Job A for CUI X
Errors returning from Job A for CUI X

Errors returning from Job B to A for CUI X
Safer Chaos Engineering
Safer Chaos engineering

Faults targeting
- On all requests
- On random requests

Risks (blast radius)
Safer Chaos engineering

Faults targeting
- On all requests
- On random requests
- On requests marked for a CUI

Risks (blast radius)
Safer Chaos engineering

Faults targeting

- On all requests
- On random requests
- On requests marked for a CUI
- On copied requests marked for a CUI and "experiment"

Risks (blast radius)
Beyond distributed tracing

**Simple idea**

- **Propagate** a piece of metadata (e.g. CUI) w/ request
- **Gather** system performance metrics by the propagating metadata (and the peer)

**Easily adoptable**

- Baggage mechanism in distributed tracing solutions
- [https://www.w3.org/TR/baggage/](https://www.w3.org/TR/baggage/)
Thank you!

Beyond Distributed Tracing

Kyusoon Lee
qsoonlee@google.com