Architectures that Scale Deep: Regaining Control in Deep Systems

Ben Sigelman (@el_bhs, bhs@lightstep.com)
Co-founder & CEO: LightStep
Co-creator: OpenTracing, OpenTelemetry, Google Dapper, Google Monarch

QCon SF, November 2019
Part I
Scaling, and Deep Systems
What is scale, anyway?
Scaling wide
Scaling wide
Scaling wide
Scaling wide
Scaling wide
Scaling deep
Scaling deep
Scaling deep
Scaling deep
Scaling deep
How does this look for software?
Software: Scaling **wide**
Software: Scaling deep
How do real-world systems look?
Microservices at scale aren’t just wide systems, they’re deep systems.
Deep Systems

Architectures with ≥ 4 layers of independently operated services
(including external/cloud dependencies)
What do deep systems sound like?
What do deep systems sound like?

“Don’t deploy on Fridays”
“Where’s Chris?! I’m dealing with a P0 and they’re the only one who knows how to debug this.”
What do deep systems sound like?

“It can’t be our fault, our dashboard says we’re healthy”
What do deep systems sound like?

“Kafka is on fire”
What do deep systems sound like?

“I need 100% availability from your team. One hundred percent.”
What do deep systems sound like?

“I didn’t know I depended on that region”
What do deep systems sound like?

“That was on a dashboard but I can’t find it”
What do deep systems sound like?

Lots of challenges:
- People-management
- Security
- Multi-tenancy
  - “Big-customer” success
- Performance
- Observability
Part II
Control Theory: TL;DR Edition
Why do we care so much about observability, anyway?
A System

... and its state vector,
Observability

A System

... and its state vector, $\vec{u}$

Inputs

Outputs

How well can you infer internal state using only the outputs?
Controllability

How well can you control internal state using only the inputs?
Controllability is the dual of Observability.
Part III
What Deep Systems Mean for Observability
Stress (n): responsibility without control

what you can control

what you are responsible for
Service-Owner Responsibility \( \approx 0 (d^2) \)

Service-Owner Control \( \approx 0 (1) \)

System Depth: \( d \)
Service-Owner Responsibility $\approx 0\ (d^2)$

Service-Owner Control $\approx 0\ (1)$

System Depth: $d$

Observability: Shrink This Gap
Mental models

A System
Managing Deep Systems

Services must have **SLOs**
(“Service Level Objectives”: latency, errors, etc)

For effective service management, only three things matter:

0. Releasing service functionality
1. Gradually improving SLOs
2. Rapidly restoring SLOs

In a deep system, we must **control the entire “triangle”** to maintain our SLOs
There’s that word again…

Controllability == Observability
Observability: “The Conventional Wisdom”

Observing microservices is hard

Google and Facebook solved this (right???)

They used **Metrics, Logging, and Distributed Tracing**…

… So we should, too.
3 Pillars, 3 Experiences

- **Metrics**
- **Logs**
- **Traces**
Three Pillars? Two giant pipes…

Without Traces:
Cognitive Load $\approx O(depth^2)$

Metrics

Logs
Three Pillars? Two giant pipes…
Two giant pipes...

Without Traces:
Cognitive Load $\approx O(\text{depth}^2)$
Traces
Traces provide Context
Traces provide context and context rules out invalid hypotheses.
Two giant pipes and a filter

Metrics

Context
(from traces)

Logs
Context reduces cognitive load

With Traces:
Cognitive Load $\approx O(depth)$
Observability: Shrink This Gap

Service-Owner Responsibility \(\approx O(d^2)\)

Service-Owner Control \(\approx O(1)\)

System Depth: \(d\)
Let’s Review
Microservices don’t just scale wide, they scale *deep*

Recognize deep systems
Stress (n): responsibility without control

what you can control

what you are responsible for
“Controllability” (of SLOs) depends on observability
“The Three Pillars of Observability” is a lousy metaphor

... and traces are not sprinkles
Tracing can reduce cognitive load from $O(depth^2)$ to $O(depth)$
Tracing is the backbone of simple observability in deep systems.
Thank You

Play with LightStep, for free, anytime:
(no email address required!)
lightstep.com/play

Feedback always welcome:
twitter → @el_bhs
the emails → bhs@lightstep.com