Practicalities of Productionizing Distributed Systems, 2018

Jeff Hodges - @jmhodges

2018-04-25

Practicalities of Productionizing Distributed Systems

Jeff Hodges
@jmhodges (2013-10-29)

Why you should listen to me

Scale-invariant

Building and running distributed systems

Quick foundation

What makes distributed systems different

Failure

A subset of failures

Garbage collection pauses

Clients stuck to an overloaded process

Socket write succeeds
here,
but fails over there

Partial failure

“It’s slow” is the hardest problem you’ll ever debug

Create partial availability

Search

“Who to Follow” in the monorail

Knowing what the system has done

Metrics are very good

Percentiles, not averages

Tracing

On profiling

Releases should change a metric

Free-form logs are liars

Common “problems” are overlogged

Uncommon problems are buried

Uncommon problems are not logged

Avoid coordination

If your problem fits in memory, it’s probably trivial

Backpressure

Dropping new messages on the floor

Returning “overload” error responses

Timeouts and exponential back-offs

Separating deploy from release

Roll out infrastructure with feature flags

```
if (Decider.available..) {
```

Multiple versions are the norm

Datacenter schedulers are worth it

“““thought leadership”””

Collaboration is politics

No time-traveling stalkers

Encryption is a moral necessity

Data minimization is a moral necessity

Okay, that’s it

Links

- https://twitter.com/jmhodges
- "Fallacies of Distributed Computing Explained" - http://www.rgoarchitects.com/Files/fallacies.pdf
- Jeff Dean's "Numbers Everyone Should Know" slide - http://www.cs.cornell.edu/projects/ladis2009/talks/dean-keynote-ladis2009.pdf
- Coda Hale's "Metrics, Metrics Everywhere" - http://www.youtube.com/watch?v=czes-oa0yik
- Basis for this presentation - http://www.somethingsimilar.com/2013/01/14/notes-on-distributed-systems-for-young-bloods/