Distributed Systems in Production

Jeff Hodges
2014-04

Why you should listen to me

Why you shouldn't listen to me

Scale-invariant

Building and running Distributed Systems

Quick foundation

What Makes Distributed Systems Different

Failure

A subset of failures

- Garbage collection spiral on a single machine causes requests to timeout
- A process is overloaded, so too many clients get stuck trying to connect to it, so it gets slower
- Socket write succeeds locally, but fails on the remote machine

Partial Failure

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

Metrics are the only way to get your job done.

On profiling

Deploys should change a metric

Logs are liars

- Common “problems” are overlogged
- Uncommon ones are hidden
- Uncommon ones are not logged

Avoid coordination

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

Back-pressure

Some Examples

- Dropping new messages on the floor
- Returning documented overload errors until the system clears
- Timeouts and exponential back-offs

Create
partial availability

Search

Who to Follow in the monorail

Consider a private messaging database

Separating deploy from release

Roll out infrastructure with feature flags

Slow, dark rollouts

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

Multiple versions are the norm

Exploit data-locality

Locality in Time, Too

- Batching
- Collapsed forwarding

Computers are Fast

Thousands of requests per sec per machine in hundreds milliseconds. Easily.

Extract services

Stricter boundaries means even less cheating.

Pulling out a service makes deploys easier

Avoids human coordination costs that libraries require.

SOA through standardization

On-call rotations

The Notorious E.O.C.

Increasing the size of my thought leadership

Robust distributed systems cost more than undistributed systems.

Robust open source distributed systems are less common

Collaboration is politics

Okay, that's it

Links

- http://go/classfeedback
- “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/