WiXEngineering

Monitoring Driven Debugging

Anna Tsibulskaya | System @ Wix.com

anna@wix.com

01

What is debugging and commonly used techniques

Debugging is the process of finding and resolving defects or problems within a computer program or a system.

Debugging techniques*:

- Print debugging
- Remote debugging
- Post mortem debugging
- “Wolf fence” algorithm

Debugging by tests (Intrusive debugging)

02

The problem of debugging issues in large-scale systems

“

Debugging using tests

is problematic to scale

for large-scale and complex systems.

Let’s look at a real case:

- 7% of all Wix RPC calls failing with *‘exceptions.RpcTransportException’*
- Occurs in all microservices (over 1000)
- Happens on all servers (over 1000)

We wrote a test to reproduce this issue:

A script that would send RPC requests to a microservice and log all failures.

We got 0 exceptions logged after running this script for several hours.

So let’s attempt to estimate the number of tests to reproduce and debug the issue:

- 1000 microservices - each makes at least 2 requests and gets 2 requests

  = 4000 calls.

- 1000 servers - each running multiple services that we much check

  = 4M calls.

- 7% failure rate - we must make at least 15 requests in each test case to

  achieve close to 100% probability of a failed request = 60M requests.

“

This is definitely not going to take 5 minutes.

We need to change our debugging approach

03

Monitoring Driven Debugging

Monitoring driven debugging is an approach that allows debugging the issue without the need to reproduce it.

In the case we discussed before:

- The issue was in changed rights on resolv.conf file (user which runs microservices wasn't able to read it, so no DNS resolving for applications)
- Using tests we found the issue in 2.5 days
- We added monitoring on DNS resolving from microservices. Next time the issue happened - it took 2 minutes to debug and 3 more to fix.

Monitoring driven debugging:

Pros:

- Fast and precise
- Doesn’t require tests
- Allows to add alerts
- Ability to add auto remediation
- Good for big and complex systems

Cons:

- Price of monitoring
- Too much monitoring data can be confusing
- Long curve to setup monitoring to be useful in debugging

More than that:

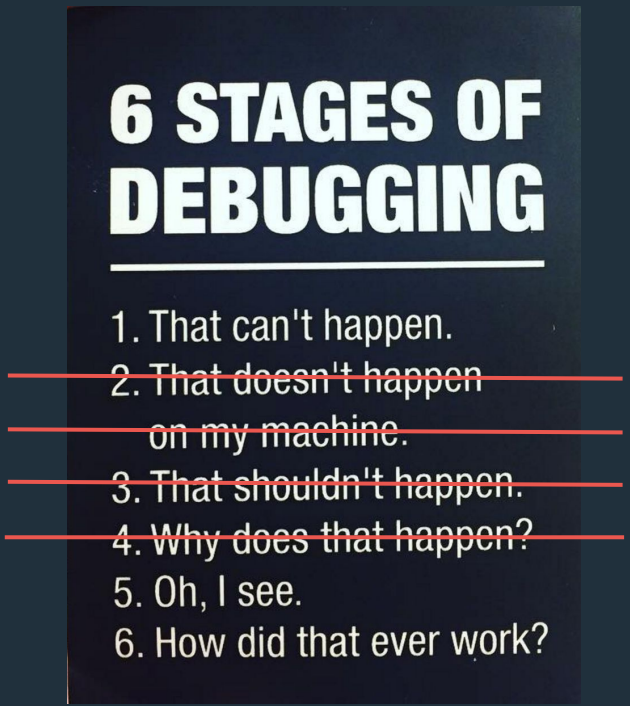
Reproducing issues in test cases requires a solid

hypothesis on the cause of the issue.

Monitoring only need to **capture the symptoms.**

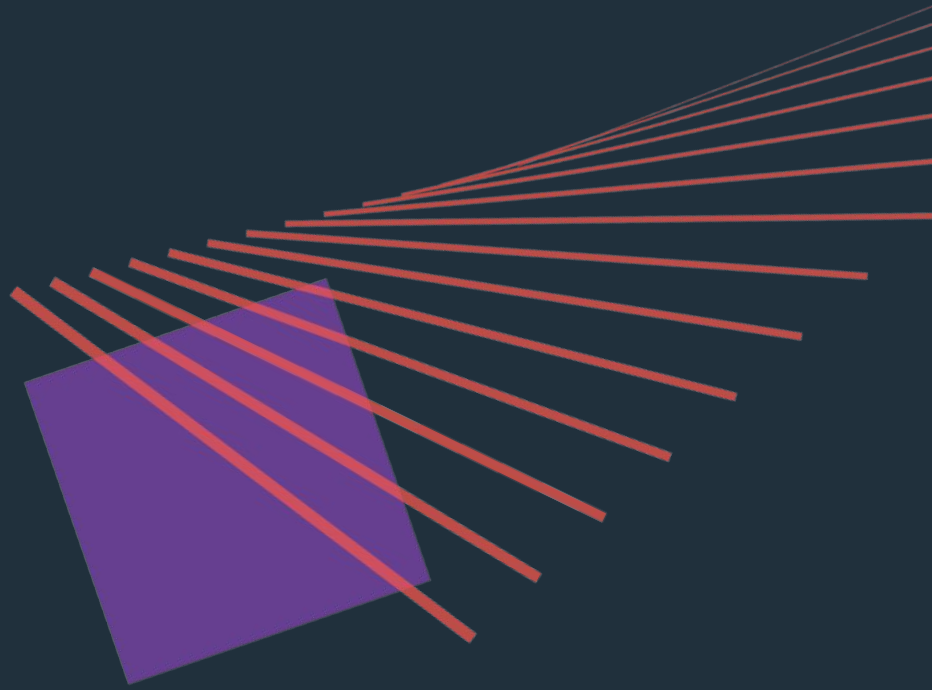
Monitoring Driven Debugging:

Just because 6 stages become 3.



WiX Engineering

Thank You



anna@wix.com