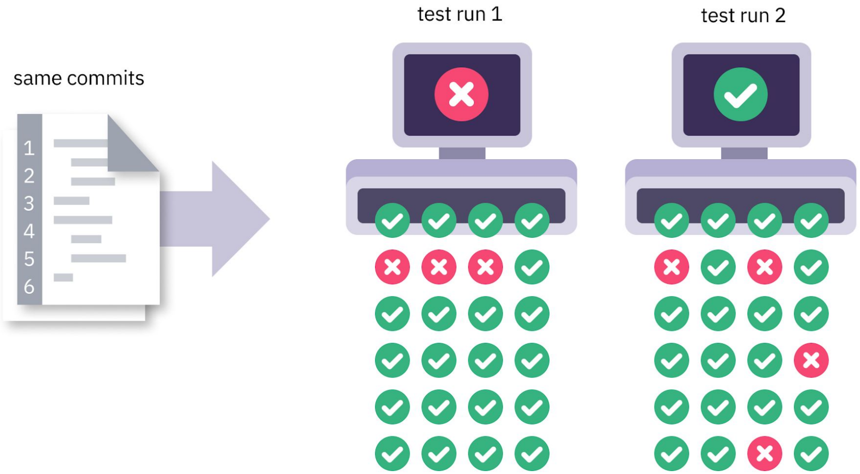
Flaky Tests at Google and How We Mitigate Them

Patrick, Jacob, Brian, Nathan, Mehak, Ethan

What are flaky tests?

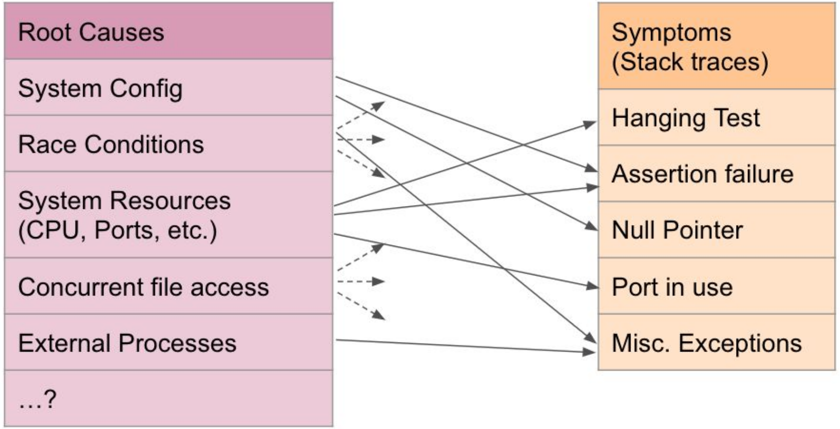
- Google defines a "flaky" test as test that exhibit both a passing and a failing result with the same code
  - At Google 1.5% of test are reported as a “flaky” test and about 16% of test have some level of “flakiness”



https://cdn.spark.app/media/launchable/image/flaky_tests_2_.png

Common Causes

- Concurrency
- Non-deterministic or undefined behaviors
- Flaky third party code
- Infrastructure problems



Root Causes

System Config

Race Conditions

System Resources (CPU, Ports, etc.)

Concurrent file access

External Processes

…?

Symptoms (Stack traces)

Hanging Test

Assertion failure

Null Pointer

Port in use

Misc. Exceptions

https://1fykyq3mdn5r21tpna3wkdyi-wpengine.netdna-ssl.com/wp-content/uploads/2021/06/Screen-Shot-2021-01-11-at-3.07.37-PM.png

Consequences

- Extra time and effort is needed to review test that change from passing to failing
    - This work is also repetitive meaning the test usually have to be reviewed multiple times
- Legitimate failures that flaky test report are often ignored which results in issues down the road
- Test have to be run multiple times which increases the amount of time it takes to find and fix legitimate failures

https://engineering.gusto.com/eliminating-flaky-ruby-tests/

Other Companies

- This issue with flaky tests are a widespread issue, not just at Google
- Companies such as Uber, and Spotify have created blog posts detailing their experiences with flaky tests

https://newsroom.spotify.com/media-kit/logo-and-brand-assets/

https://techcrunch.com/2018/09/12/uber-is-getting-a-new-look/

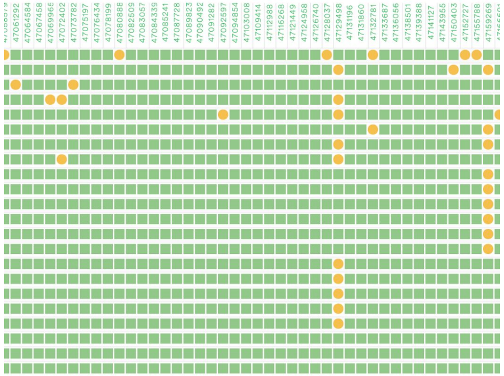
Uber

- Separate pipeline for flaky tests
  - If a test is run multiple times and passes once and fails more than once, consider it flaky and it gets moved into the flaky test pipeline
  - By default, flaky tests are ignored and not used when testing new code is being merged
- Try to identify what causes the tests to be flaky
  - When a test is determined to be flaky, run automated tools to determine if a port collision is causing the flaky test
  - Have a “fix it week”
  - Encourage teams with the highest levels of flaky tests to work on their tests
  - Static code analysis

https://eng.uber.com/handling-flaky-tests-java/

Spotify

- Test result visualization with Odeneye
- Table that shows developers the time each test takes, and how flaky it is, which helped reduce test flakiness from 6%-4%
- Created a git bot that developers could use to run a flaky test multiple times to make sure that the test is no longer flaky once they have fixed it



https://engineering.atspotify.com/2019/11/test-flakiness-methods-for-identifying-and-dealing-with-flaky-tests/

Mitigation Strategies

- The ability to only re-run test that fail and to only report an issue if a test fails 3 times in a row
- A tool that monitors flakiness and quarantines test when the flakiness becomes too high
- A tool that monitors changes in flakiness levels of test and tries to find the cause of the change
- A team dedicated to providing accurate and timely information about test flakiness

How can programmers avoid writing flaky tests in the first place?