

NEED FOR SPEED

accelerate tests from 3 hours to 3 minutes



emo@komfo.com



With slow tests you're
shipping crap faster.

Everyone Loves Unit Tests



High Level Tests Problems

The tests are slow

The tests are unreliable

The tests can't exactly pinpoint the problem

3

hours

600 API tests

3

minutes

Before

	#1996	3 hr 6 min	slave1
	#1995	3 hr 1 min	slave1
	#1994	3 hr 7 min	slave1
	#1993	3 hr 10 min	slave1
	#1992	3 hr 4 min	slave1
	#1991	3 hr 6 min	slave1
	#1990	3 hr 11 min	slave1
	#1989	3 hr 6 min	slave1
	#1988	3 hr 12 min	slave1
	#1987	3 hr 5 min	slave1
	#1986	3 hr 43 min	slave1
	#1985	4 hr 40 min	slave1

After

	#1743	2 min 53 sec	slave1
	#1742	2 min 53 sec	slave1
	#1741	2 min 54 sec	slave1
	#1740	2 min 53 sec	slave1
	#1739	3 min 6 sec	slave1
	#1738	3 min 0 sec	slave1
	#1737	2 min 55 sec	slave1
	#1736	2 min 52 sec	slave1
	#1735	2 min 54 sec	slave1
	#1734	2 min 54 sec	slave1
	#1733	2 min 56 sec	slave1
	#1732	3 min 11 sec	slave1

The 3 Minute Goal

**It's not about the numbers you'll see
or the techniques.**

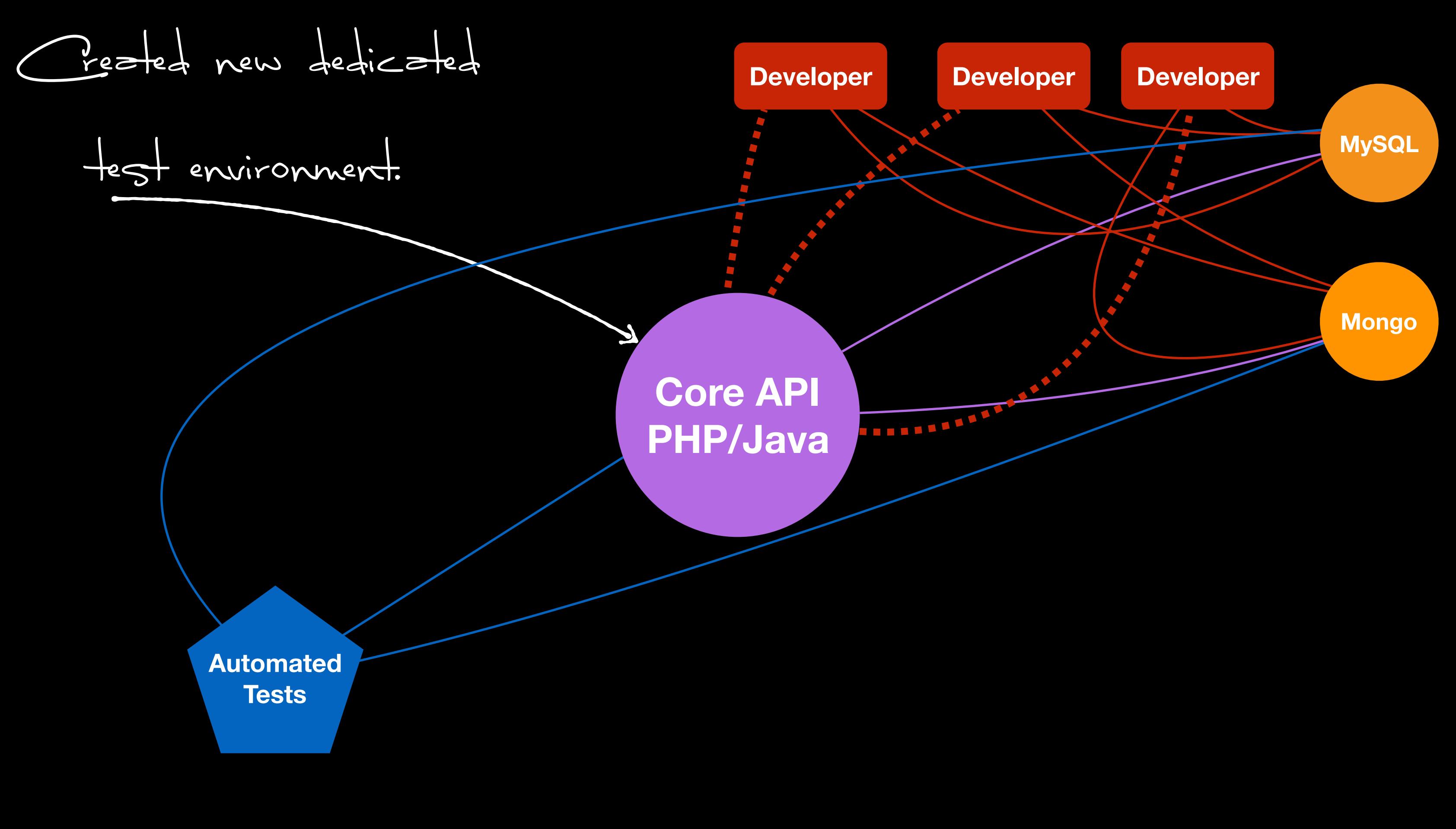
It's all about continuous improvement.

Key Steps



A photograph of a multi-lane highway with traffic moving away from the viewer. A bridge overpass is visible in the background. The road has yellow dashed lines and solid white lines. The surrounding area is green and hilly.

Dedicated
Environment



Execution Time in Minutes

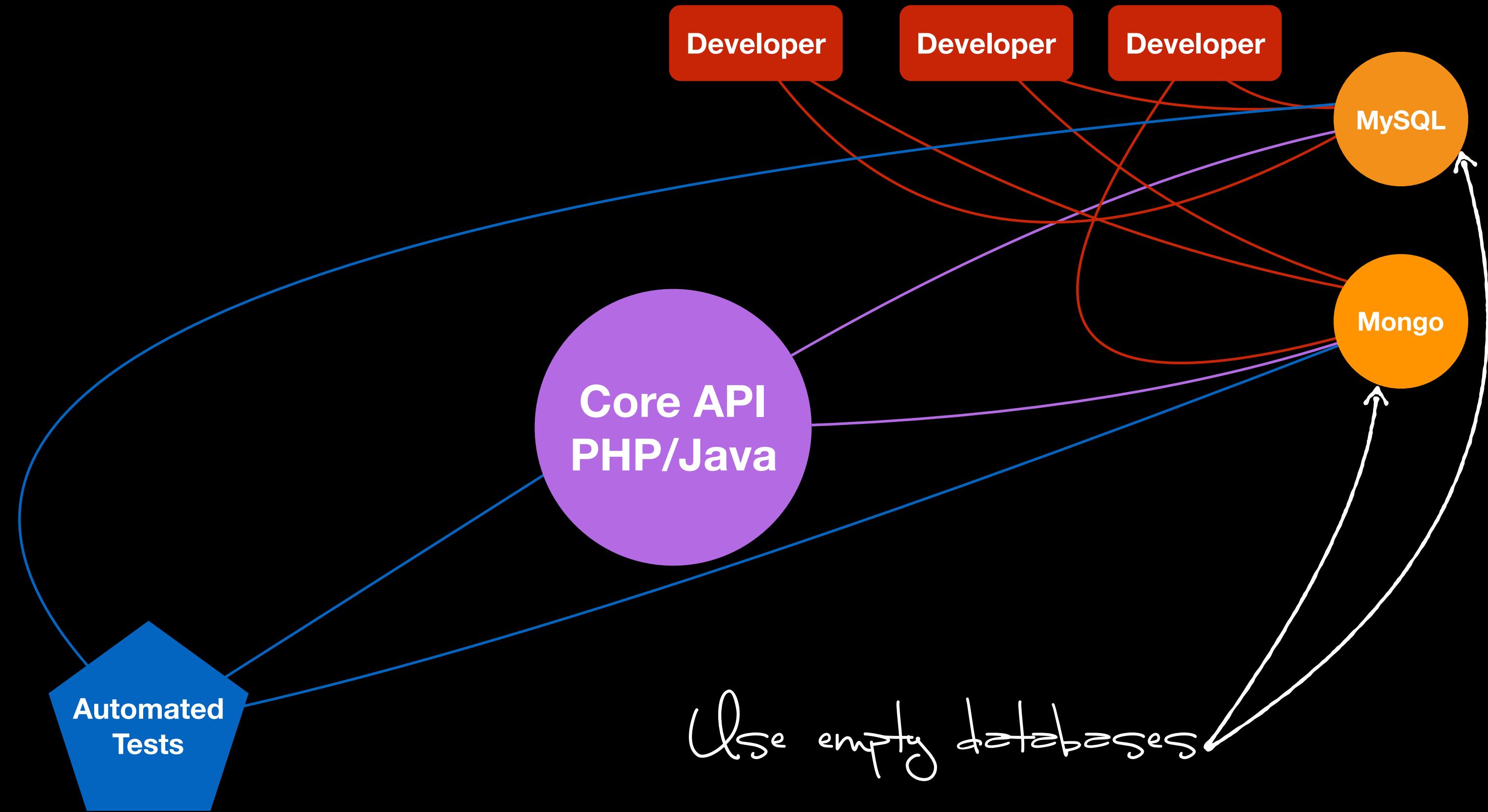
180

123

New Environment

Empty Databases

A photograph of a large auditorium or lecture hall. The seating consists of numerous rows of red plastic chairs with blue headrest covers. The chairs are arranged in a semi-circular pattern, facing towards the front of the room. The background shows the dark, curved wall of the auditorium, which appears to have some small, glowing blue lights or screens. The overall atmosphere is one of emptiness and order.



Tests need to setup all the data they need!

The time needed to create data for each test:

Call 12 API endpoints

Modify data in 11 tables

And then the test starts

Takes about 1.2 seconds

Only the DB schema and config tables (~20) are needed.

Get the current DB schema and apply it before the test run starts.

Execution Time in Minutes

180

123

89

Empty Databases

Simulate Dependencies

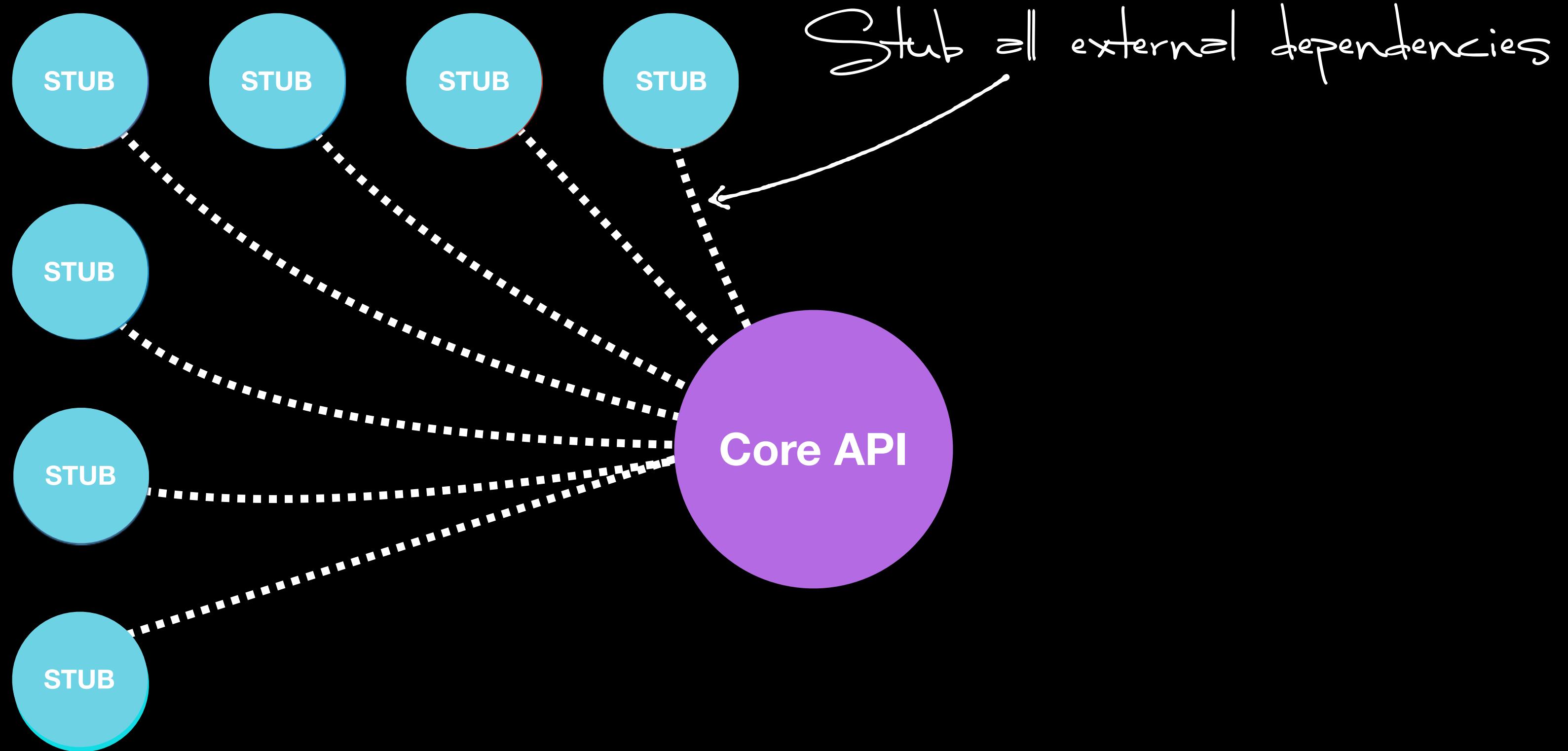


Problems with external dependencies

Sketchy Internet

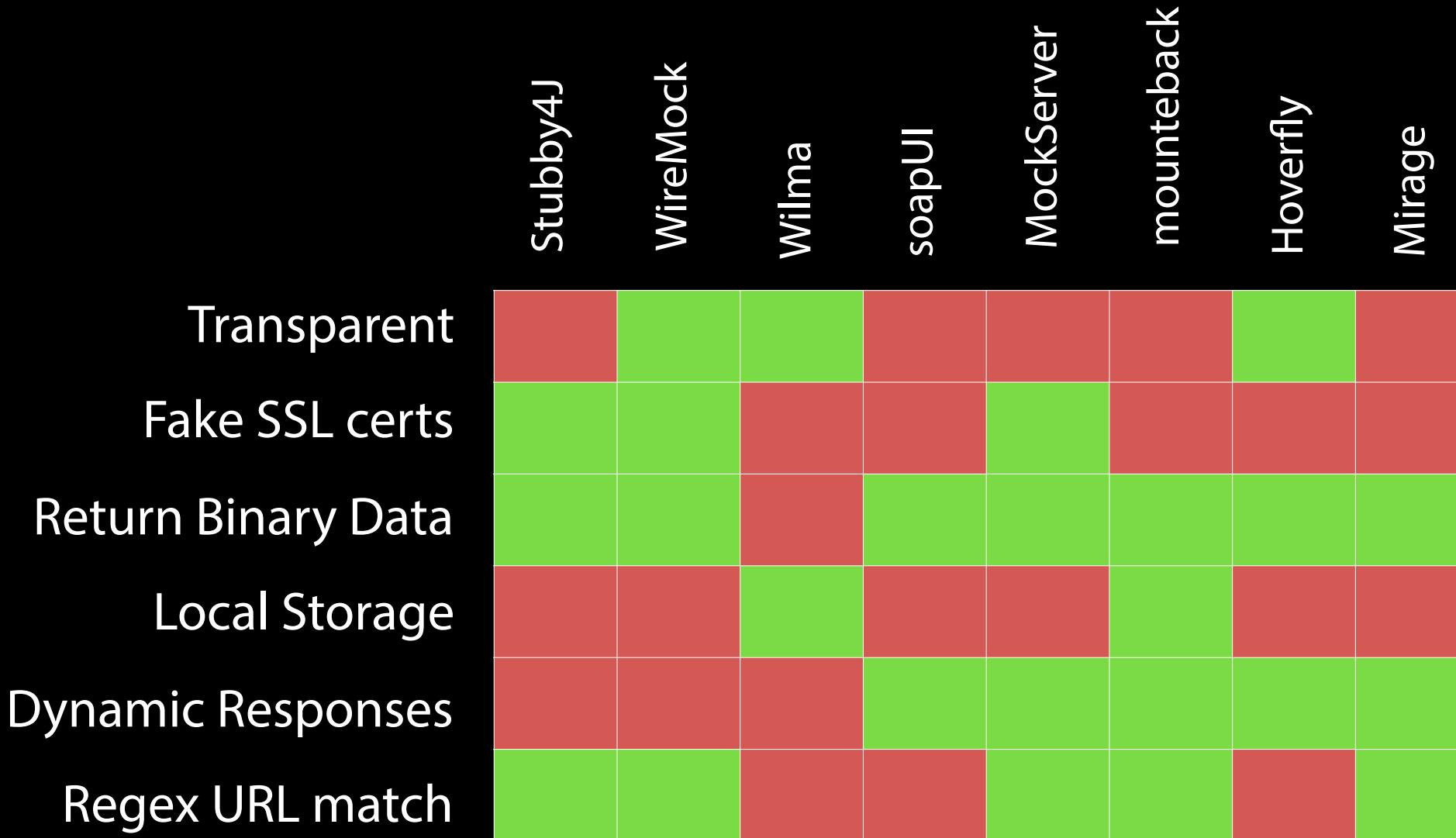
Throttling API Calls

Expiring Credentials



Stub all external dependencies

Existing Tools (March 2016)



We created project Nagual,
soon on Github.

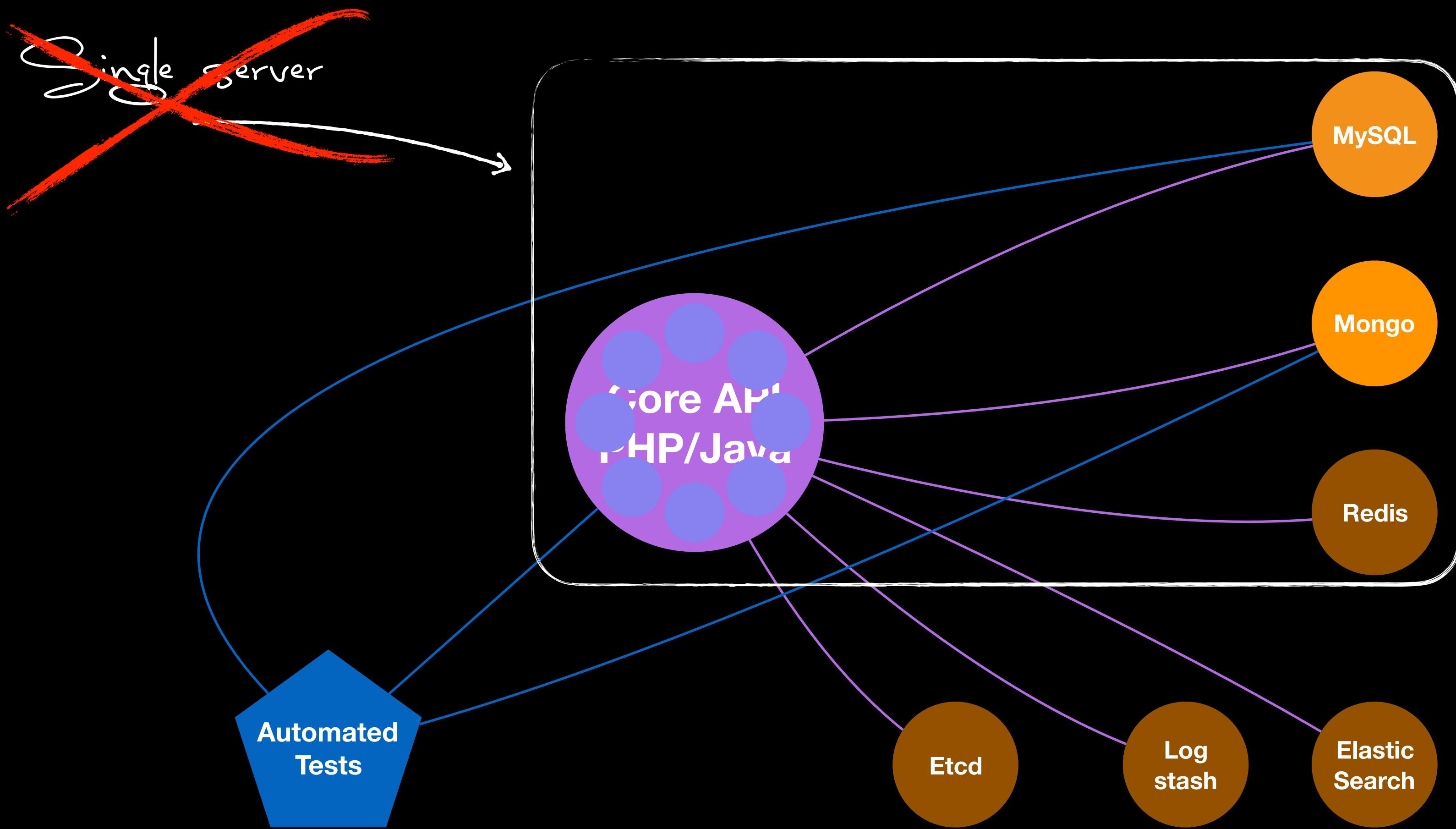
**Some of the tests still need to contact
the real world.**

Execution Time in Minutes



Move to Containers

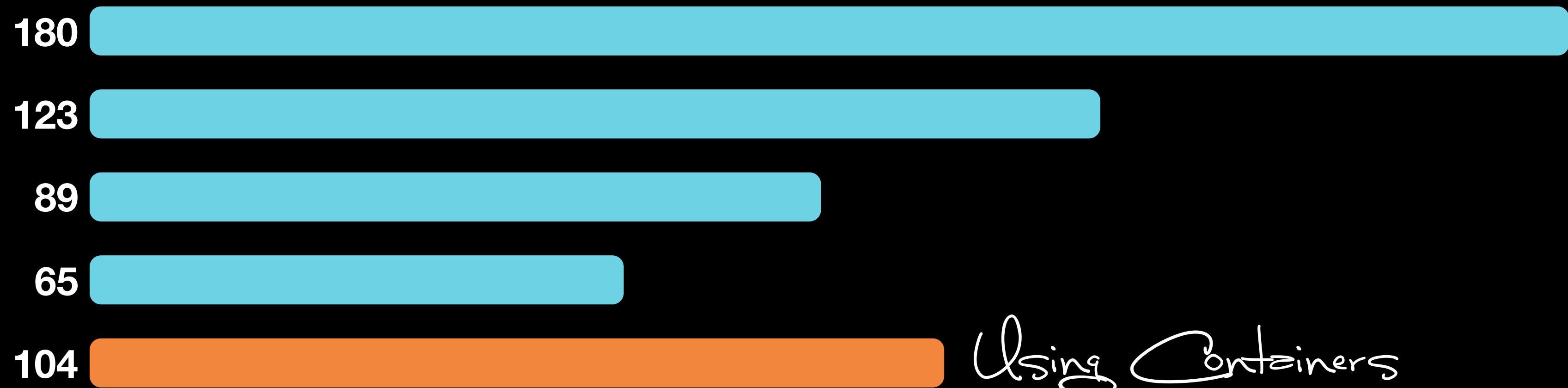




**To cope with increasing complexity
we created one container per service.**

But we we're in for a surprise!

Execution Time in Minutes





Run Databases
in Memory

Only in memory

mysqld some_options --datadir /dev/shm

Execution Time in Minutes

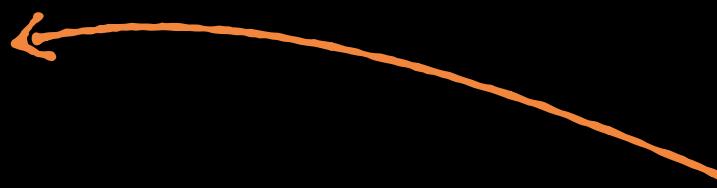


A woman in a maid's uniform, consisting of a white apron over a dark dress and a white headband, is crouching on a brick floor and mopping a heavily stained, greyish-white wall. She is looking directly at the camera with a weary expression. A bucket sits next to her on the floor. The wall behind her is covered in brown and blue stains, with the word "ROLEX" written in black spray paint. The overall scene conveys a sense of exhaustion and the futility of cleaning dirty data.

**Don't Clean
Test Data**

The cost to delete data after **every** test case

Call 4 API endpoints



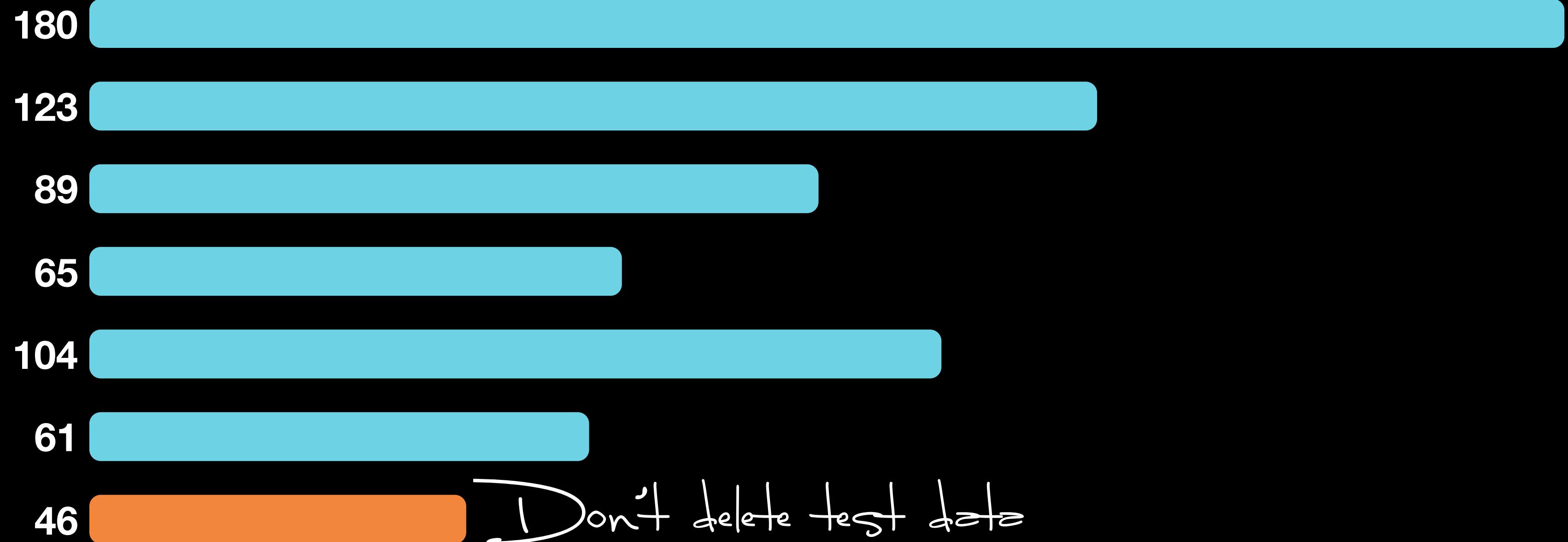
Takes about 1.5 seconds

Remove data from 23 tables



Or, stop the container, the data evaporates

Execution Time in Minutes



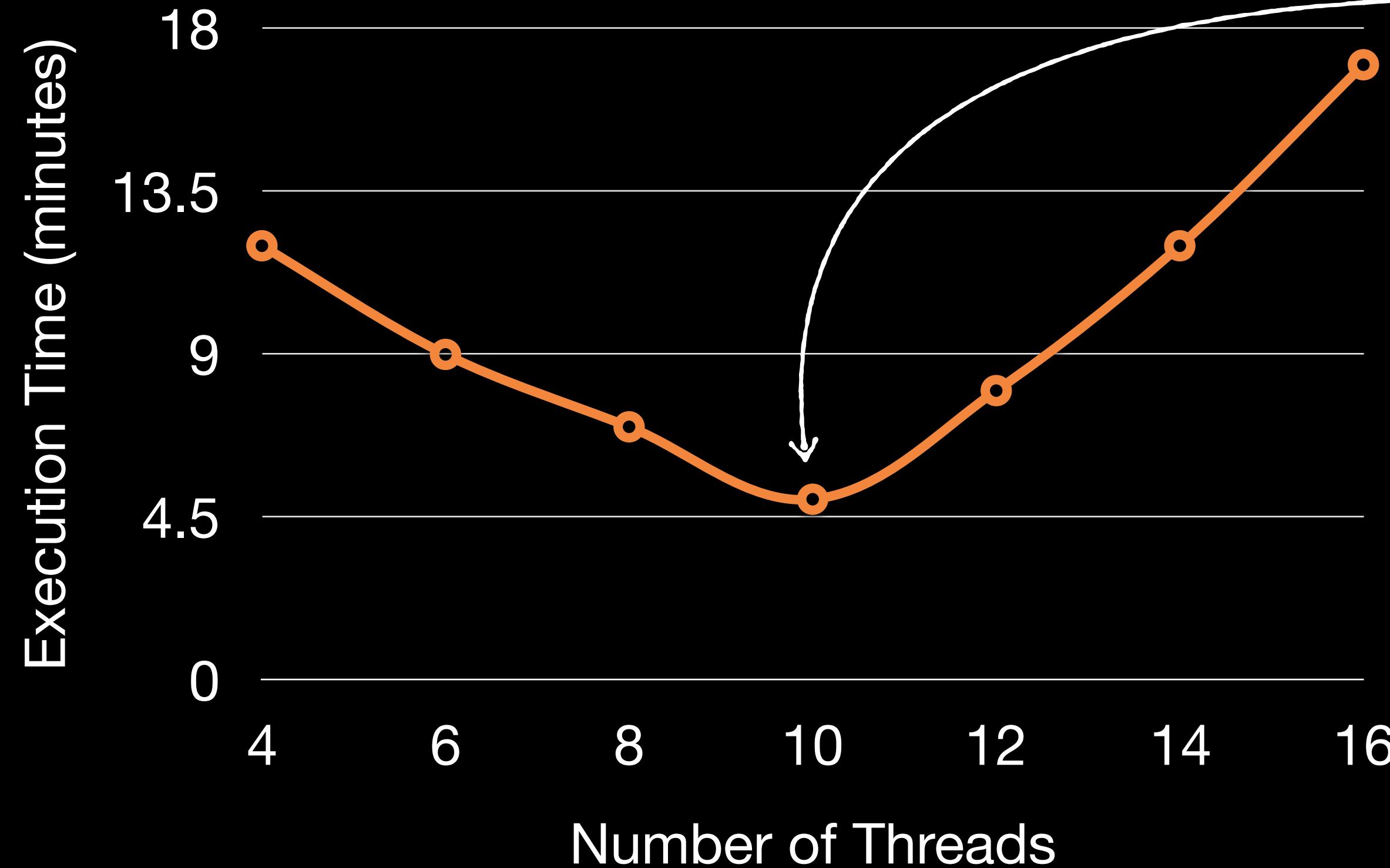


Run in Parallel

This should be your last resort, after you've exhausted all other options.

We can do this because every test creates its own test data and is independent.

The Sweet Spot



**Had to make some adjustments
to accommodate the fast tests.**

All timestamps had to be in milliseconds

```
"favorited": false,  
"retweeted": false,  
"lang": "en",  
"updated_time": ISODate("2015-10-26T14: 12: 32.715Z"),  
"gathered_time": ISODate("2015-09-25T14: 02: 32.712Z"),  
"from_user_id": NumberInt(306814990),
```

Twitter returns only seconds

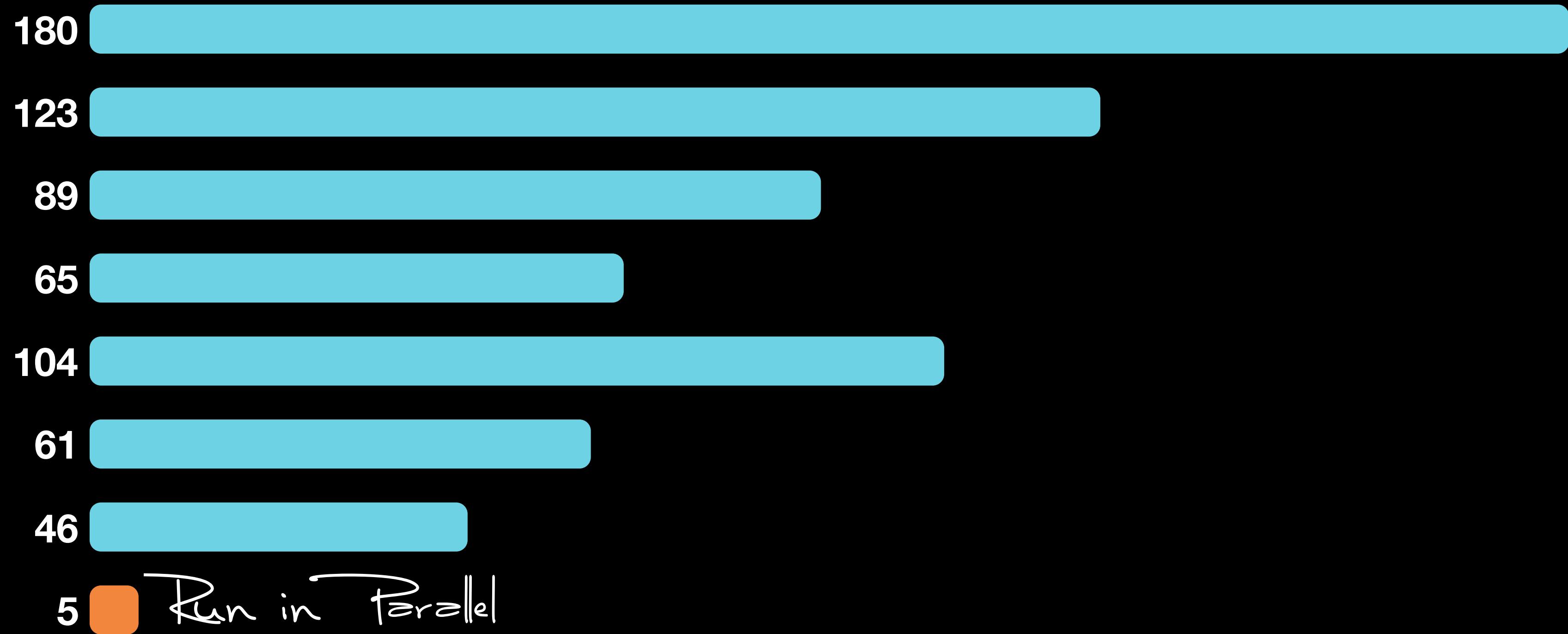
We're adding milliseconds

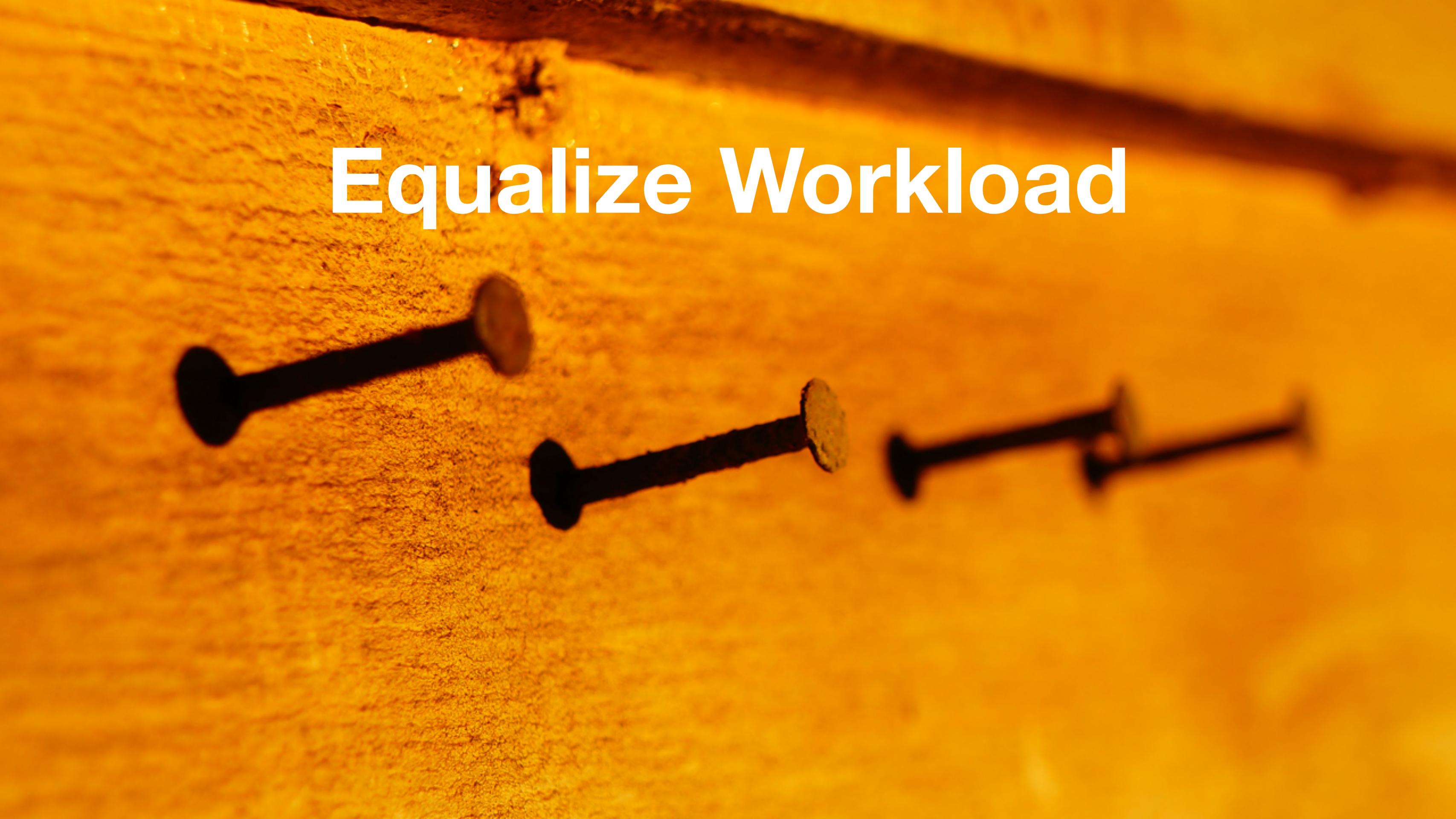
```
// too fast tests, too much deadlocks

try {
    $this->insertInTable($record);
}

catch(Exception $exception) {
    usleep(rand(100, 500));
    $this->insertInTable($record);
}
```

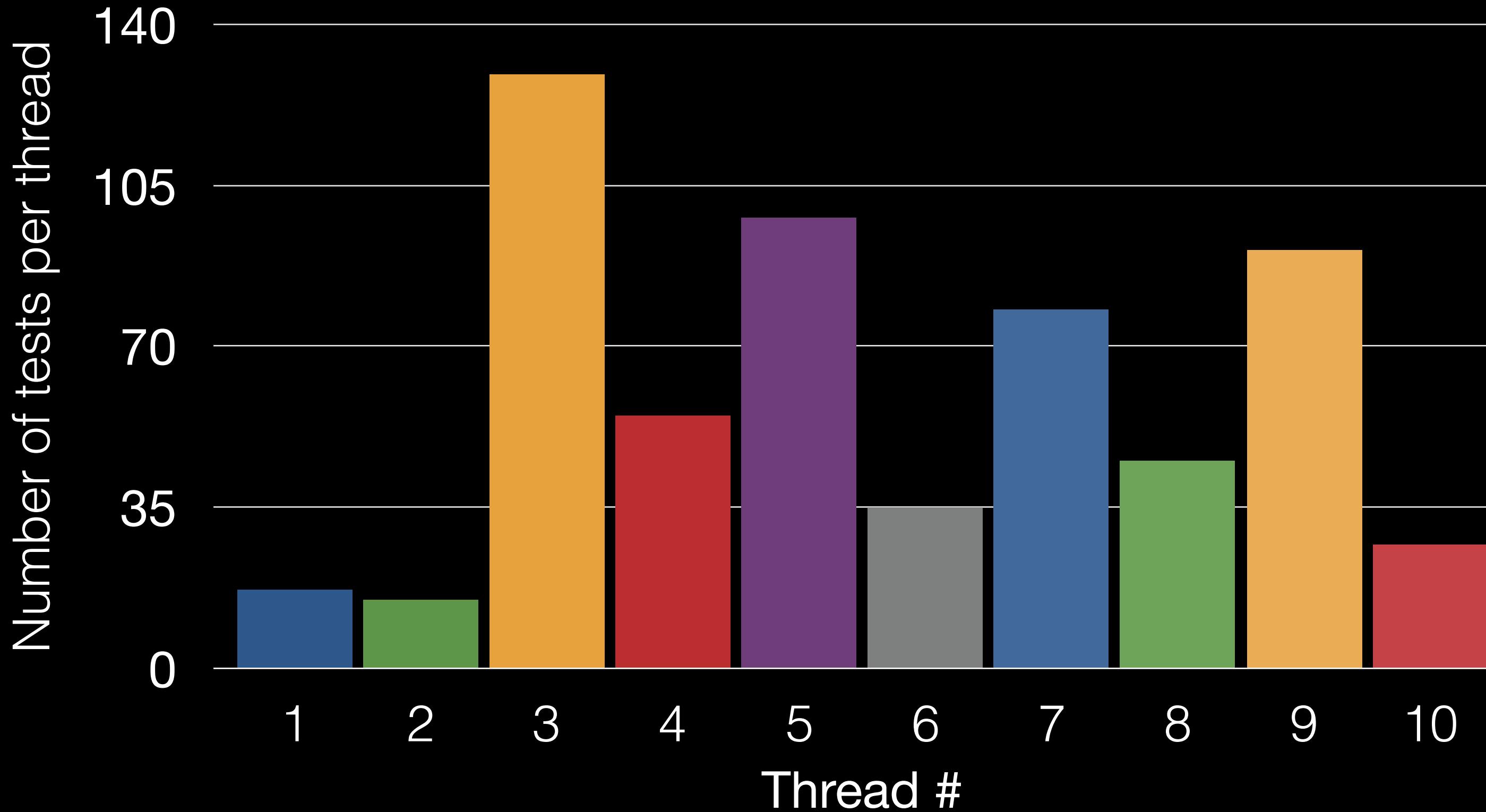
Execution Time in Minutes



A close-up photograph of a light-colored wooden surface. Three dark, metallic pins or tacks are driven into the wood at different angles. The wood has a visible grain and some minor texture. The lighting is warm, creating a golden glow.

Equalize Workload

Before



Execution Time in Minutes



The Outcome

```
590 scenarios (590 passed)  
2581 steps (2581 passed)
```

2:15 min.

Took 135.913191729 seconds

After Hardware Upgrade

```
586 scenarios (586 passed)  
2554 steps (2554 passed)
```

1:38 min.

Took 98 seconds (1:38)

High Level Tests ~~Problems~~

3 Minutes

~~The tests are slow~~

No external dependencies

~~The tests are unreliable~~

It's cheap to run all tests after every commit

~~The tests can't exactly pinpoint the problem~~

How about the UI tests?

51

minutes

12*

minutes

*Running in single thread

One more thing...

After all the tests complete, check:

Log files - errors, exceptions

Databases - wrong, unexpected data

Execution time - load, performance issues

In a couple of years, running **all your automated tests**, after every code change, **for less than 3 minutes**, will be standard development practice.

How to Start



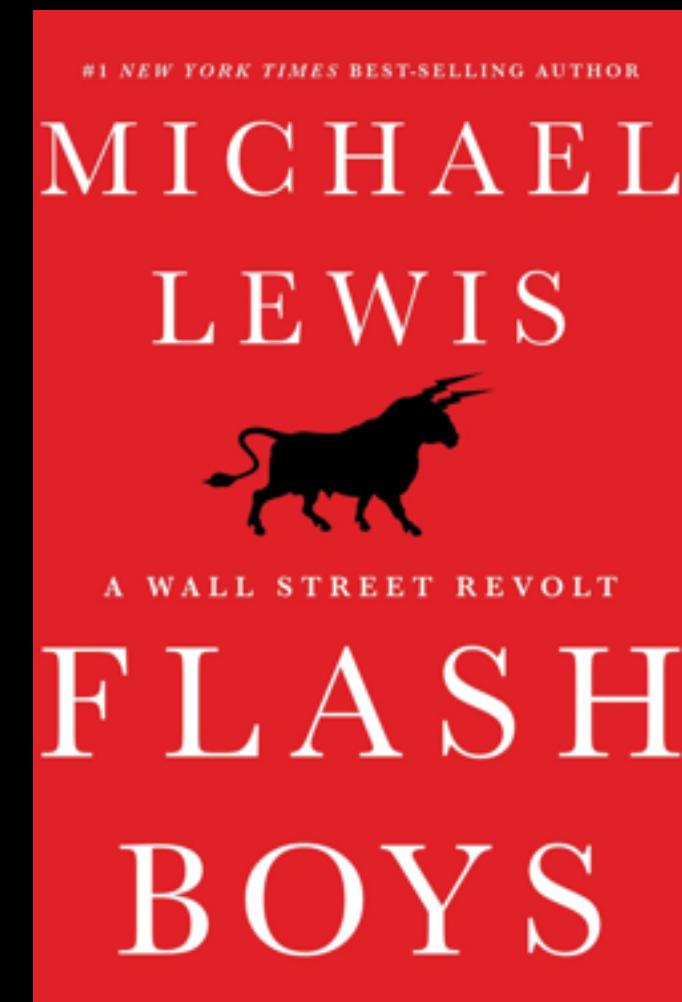
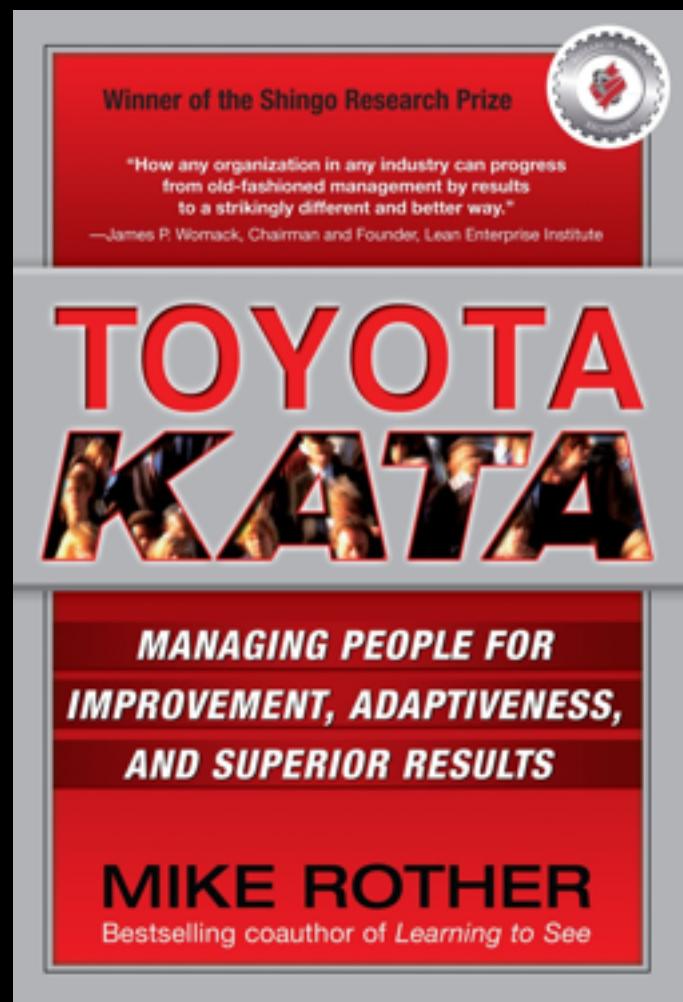
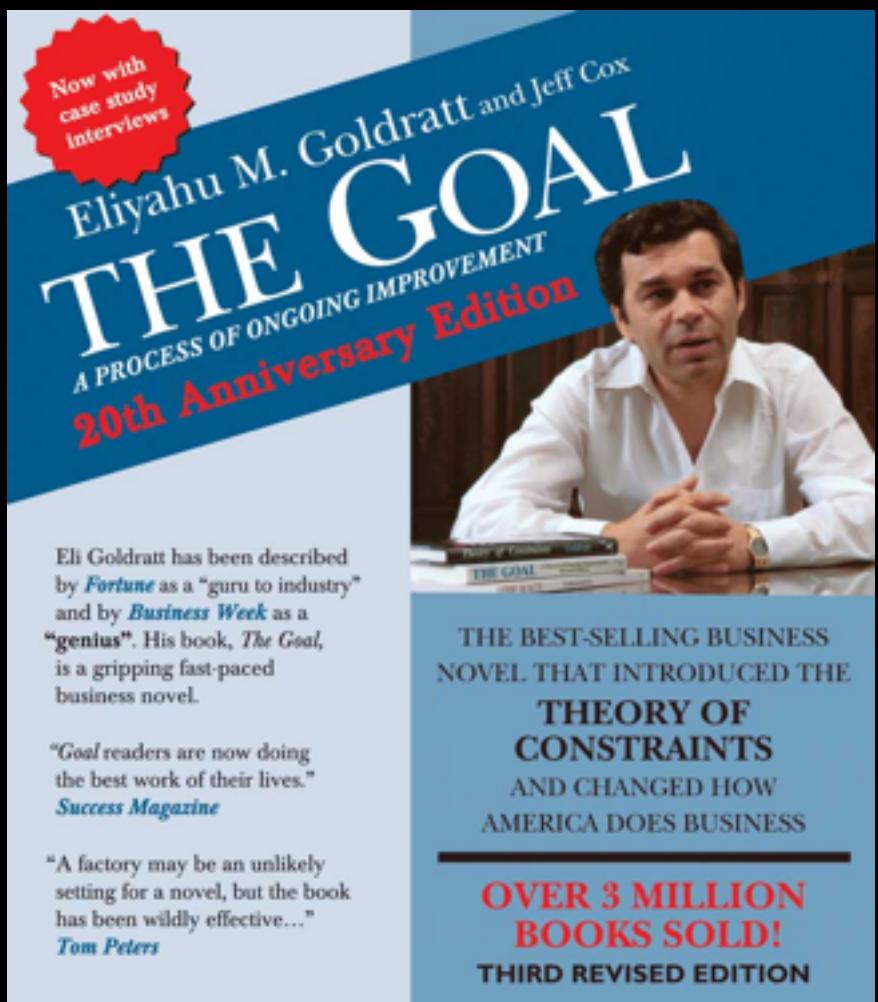
Your tests should create all the data they need

Create dedicated automation test environment

Simulate external dependencies

Run in parallel and scale horizontally

Recommended Reading



EmanuilSlavov.com

@EmanuilSlavov

AVAILABLE
BY OWNER
TAIL/OFFICE
STORE FRONT
ED 914-906-2853

HOW DO THE DEAD BUGS
GET INTO THE
ENCLOSED LIGHT FIXTURES

MEDIEVAL TACTIC

Slide #, Photo Credits

1. <https://www.flickr.com/photos/thomashawk>
2. <https://www.flickr.com/photos/paulineguilmot>
10. <https://www.flickr.com/photos/100497095@N02>
13. <https://www.flickr.com/photos/andrewmalone>
19. <https://www.flickr.com/photos/astrablog>
25. <https://www.flickr.com/photos/foilman>
29. <https://www.flickr.com/photos/missusdoubleyou>
32. <https://www.flickr.com/photos/canonsnapper>
35. <https://www.flickr.com/photos/anotherangle>
42. <https://www.flickr.com/photos/-aismist>