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
<table>
<thead>
<tr>
<th>Before</th>
<th>After</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1996</td>
<td>#1743</td>
</tr>
<tr>
<td>3 hr 6 min</td>
<td>2 min 53 sec</td>
</tr>
<tr>
<td>slave1</td>
<td>slave1</td>
</tr>
<tr>
<td>#1995</td>
<td>#1742</td>
</tr>
<tr>
<td>3 hr 1 min</td>
<td>2 min 53 sec</td>
</tr>
<tr>
<td>slave1</td>
<td>slave1</td>
</tr>
<tr>
<td>#1994</td>
<td>#1741</td>
</tr>
<tr>
<td>3 hr 7 min</td>
<td>2 min 54 sec</td>
</tr>
<tr>
<td>slave1</td>
<td>slave1</td>
</tr>
<tr>
<td>#1993</td>
<td>#1740</td>
</tr>
<tr>
<td>3 hr 10 min</td>
<td>2 min 53 sec</td>
</tr>
<tr>
<td>slave1</td>
<td>slave1</td>
</tr>
<tr>
<td>#1992</td>
<td>#1739</td>
</tr>
<tr>
<td>3 hr 4 min</td>
<td>3 min 6 sec</td>
</tr>
<tr>
<td>slave1</td>
<td>slave1</td>
</tr>
<tr>
<td>#1991</td>
<td>#1738</td>
</tr>
<tr>
<td>3 hr 6 min</td>
<td>3 min 0 sec</td>
</tr>
<tr>
<td>slave1</td>
<td>slave1</td>
</tr>
<tr>
<td>#1990</td>
<td>#1737</td>
</tr>
<tr>
<td>3 hr 11 min</td>
<td>2 min 55 sec</td>
</tr>
<tr>
<td>slave1</td>
<td>slave1</td>
</tr>
<tr>
<td>#1989</td>
<td>#1736</td>
</tr>
<tr>
<td>3 hr 6 min</td>
<td>2 min 52 sec</td>
</tr>
<tr>
<td>slave1</td>
<td>slave1</td>
</tr>
<tr>
<td>#1988</td>
<td>#1735</td>
</tr>
<tr>
<td>3 hr 12 min</td>
<td>2 min 54 sec</td>
</tr>
<tr>
<td>slave1</td>
<td>slave1</td>
</tr>
<tr>
<td>#1987</td>
<td>#1734</td>
</tr>
<tr>
<td>3 hr 5 min</td>
<td>2 min 54 sec</td>
</tr>
<tr>
<td>slave1</td>
<td>slave1</td>
</tr>
<tr>
<td>#1986</td>
<td>#1733</td>
</tr>
<tr>
<td>3 hr 43 min</td>
<td>2 min 56 sec</td>
</tr>
<tr>
<td>slave1</td>
<td>slave1</td>
</tr>
<tr>
<td>#1985</td>
<td>#1732</td>
</tr>
<tr>
<td>4 hr 40 min</td>
<td>3 min 11 sec</td>
</tr>
<tr>
<td>slave1</td>
<td>slave1</td>
</tr>
</tbody>
</table>

The 3 Minute Goal
It’s not about the numbers you’ll see or the techniques.

It’s all about continuous improvement.
Key Steps
Dedicated Environment
Created new dedicated test environment.
Empty Databases
Use empty databases
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)

<table>
<thead>
<tr>
<th>Feature</th>
<th>Stubby4J</th>
<th>WireMock</th>
<th>Wilma</th>
<th>soapUI</th>
<th>MockServer</th>
<th>mounteback</th>
<th>Hoverfly</th>
<th>Mirage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transparent</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Fake SSL certs</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Return Binary Data</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Local Storage</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Dynamic Responses</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Regex URL match</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>

We created project Nagual, soon on Github.
Some of the tests still need to contact the real world.
Execution Time in Minutes

180
123
89
65

Stub Dependencies
Move to Containers
To cope with increasing complexity we created one container per service.

But we’re in for a surprise!
Execution Time in Minutes

180
123
89
65
104

Using Containers
Run Databases in Memory
mysqld some_options  --datadir /dev/shm
Run Databases in Memory
Don’t Clean Test Data
The cost to delete data after every test case

Call 4 API endpoints

Remove data from 23 tables

Or, stop the container, the data evaporates

Takes about 1.5 seconds
Don’t delete test data
Run in Parallel
We can do this because every test creates its own test data and is independent.

This should be your last resort, after you’ve exhausted all other options.
Execution Time (minutes)

Number of Threads

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

// too fast tests, too much deadlocks
Equalize Workload
Execution Time in Minutes

180
123
89
65
104
61
46
5
3

New Environment
Empty Databases
Stub Dependencies
Using Containers
Run Databases in Memory
Don’t delete test data
Run in Parallel
Equal Batches
The Outcome

After Hardware Upgrade

590 scenarios (590 passed)
2581 steps (2581 passed)
Took 135.013191729 seconds

586 scenarios (586 passed)
2554 steps (2554 passed)
Took 98 seconds (1:38)
High Level Tests Problems

3 Minutes
No external dependencies
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.
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

- **The Goal** by Eliyahu M. Goldratt and Jeff Cox
- **Toyota Kata** by Mike Rother
- **Flash Boys** by Michael Lewis
EmanuilSlavov.com

@EmanuilSlavov
HOW DO THE DEAD BUGS GET INTO THE ENCLOSED LIGHT FIXTURES
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
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