# How to Test Asynchronous Code

*by Felix Geisendörfer*

05.05.2011 (v1)

@felixge

Twitter / GitHub / IRC

Core Contributor

&

Module Author

node-mysql

node-formidable

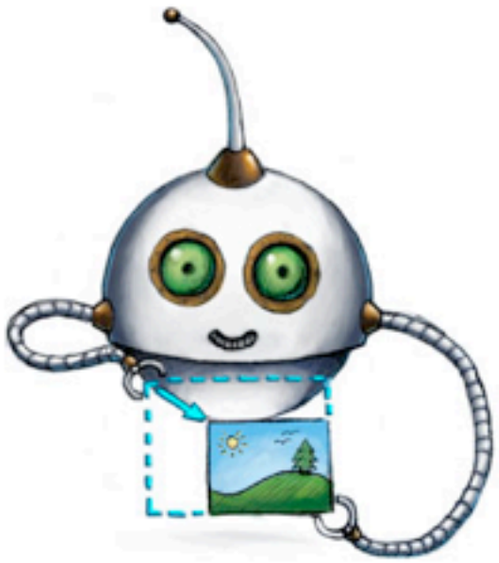
There really are only two acceptable models of development: "think and analyze" or "years and years of testing on thousands of machines".

*Linus Torvalds*

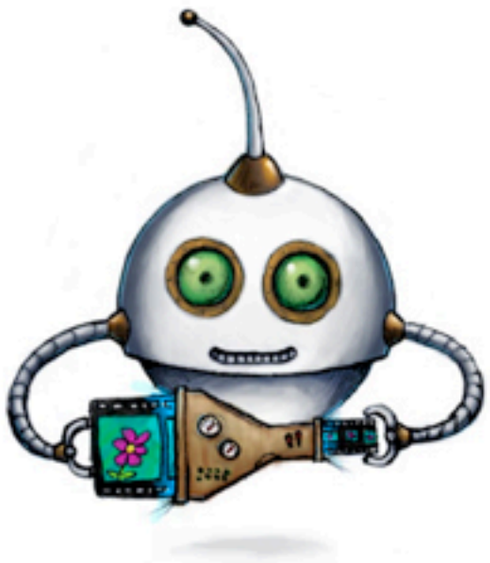
# Story Time

# ~ June 2009

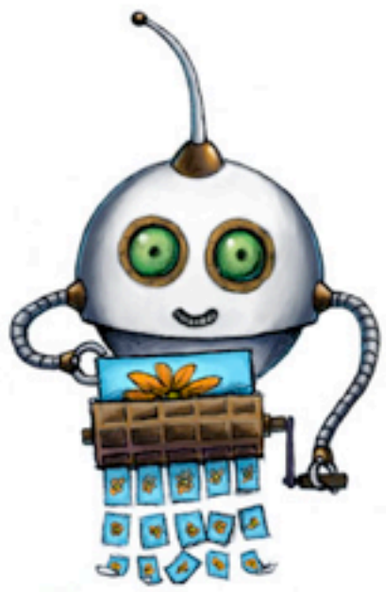
Transloadit



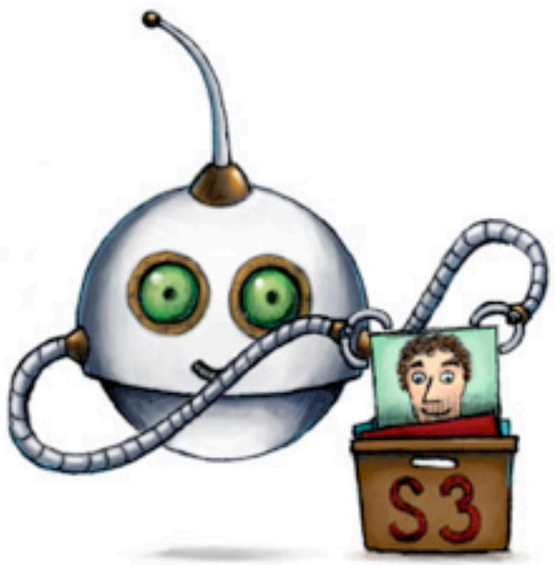
Resize images



Encode videos



Extract thumbnails




Store in S3

File uploading & processing as an infrastructure service for web & mobile applications.

We found node

node.JS™

We found node

We fell in love

We had fun

We crashed

~1.5 Years ago

# Transloadit v1

- Complete failure

- Node randomly crashed with:

```
(evcom) recv() Success
```

- A mess of code, hard to maintain

* http://transloadit.com/blog/2010/04/to-launch-or-not

# We had no tests

Time to rewrite

# How to Test Asynchronous Code

```
db.query('SELECT A', function() {
  db.query('SELECT B', function() {
    db.query('SELECT C', function() {
      db.query('SELECT D', function() {
        // WTF
      });
    });
  });
});
```

# Some old news

# How to Test Asynchronous Code

# Make it “synchronous”

# Version 2

Well, almost : )

# Transloadit v2

Now

- Processed many TB of data

- Very low bug rate

- Clean code base, ~99% test coverage

# How to Microtest Node.js Code

# Rule #1: No I/O

# Rule #2: TDD

# Rule #3: Discipline

# Libraries

# felixge/node-fake

# felixge/node-microtest

# node-microtest

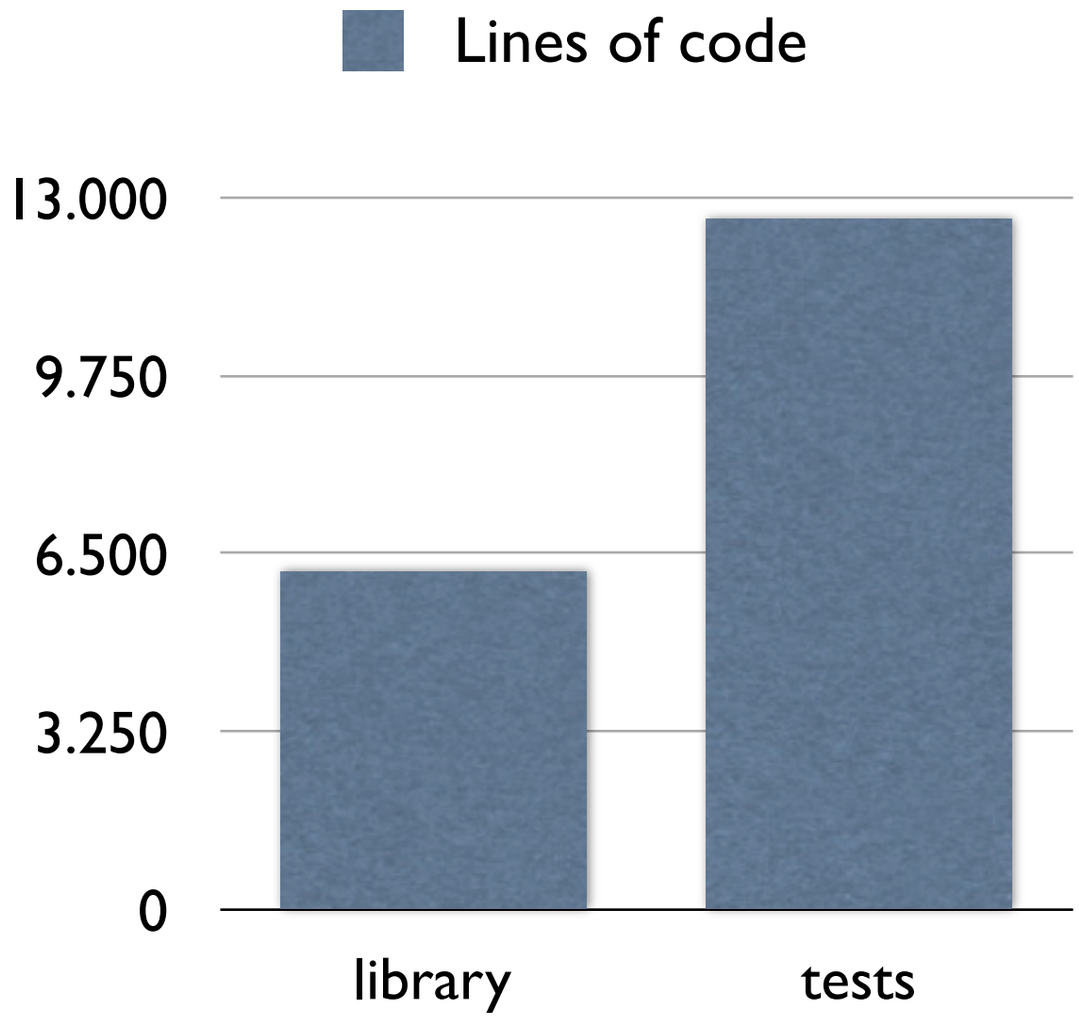
- Load test files in separate v8 context
- node-fake as mixin
- Basic test.describe(name, fn) framework

# felixge/node-far

# Numbers

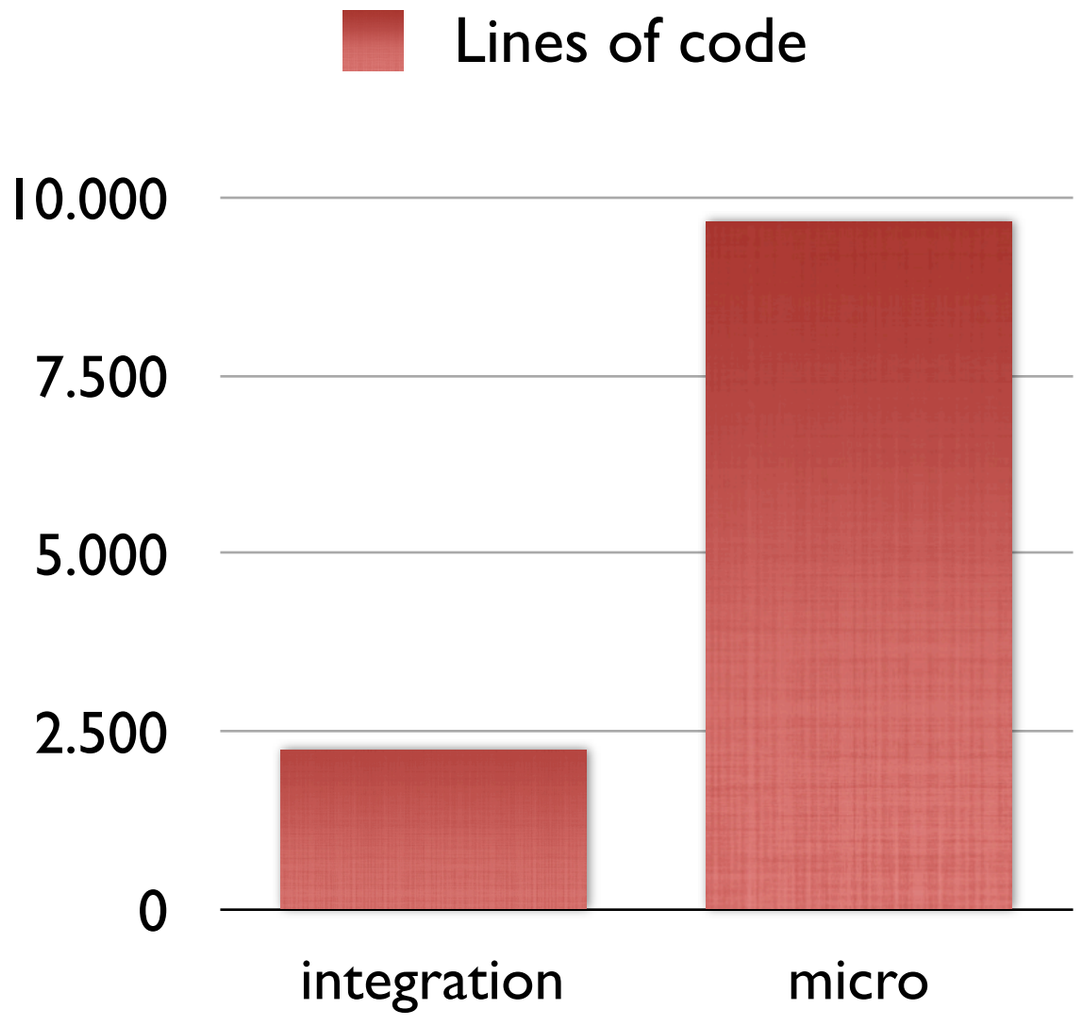
# Transloadit

# Transloadit



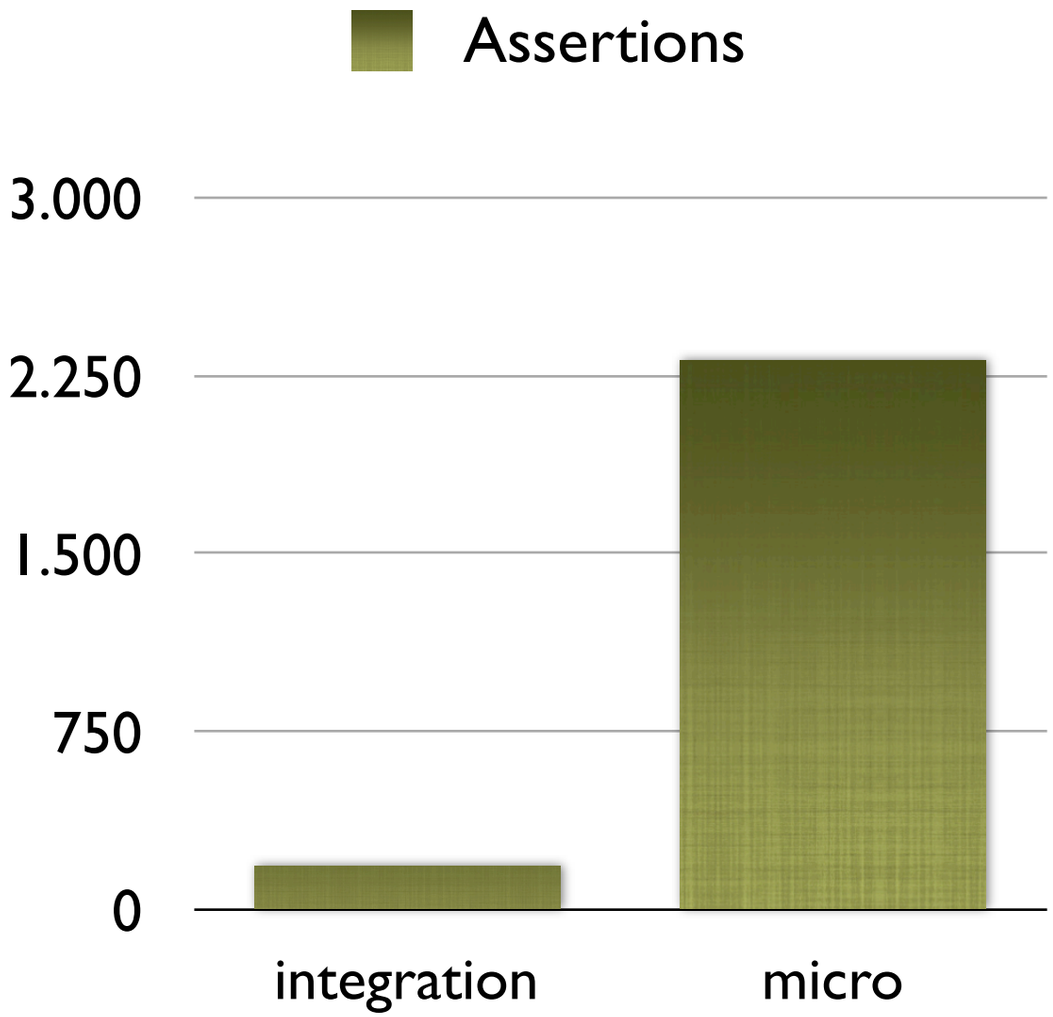

library vs test code: 1x : 2.04x

# Transloadit




integration vs. micro tests: 1x : 4.30x

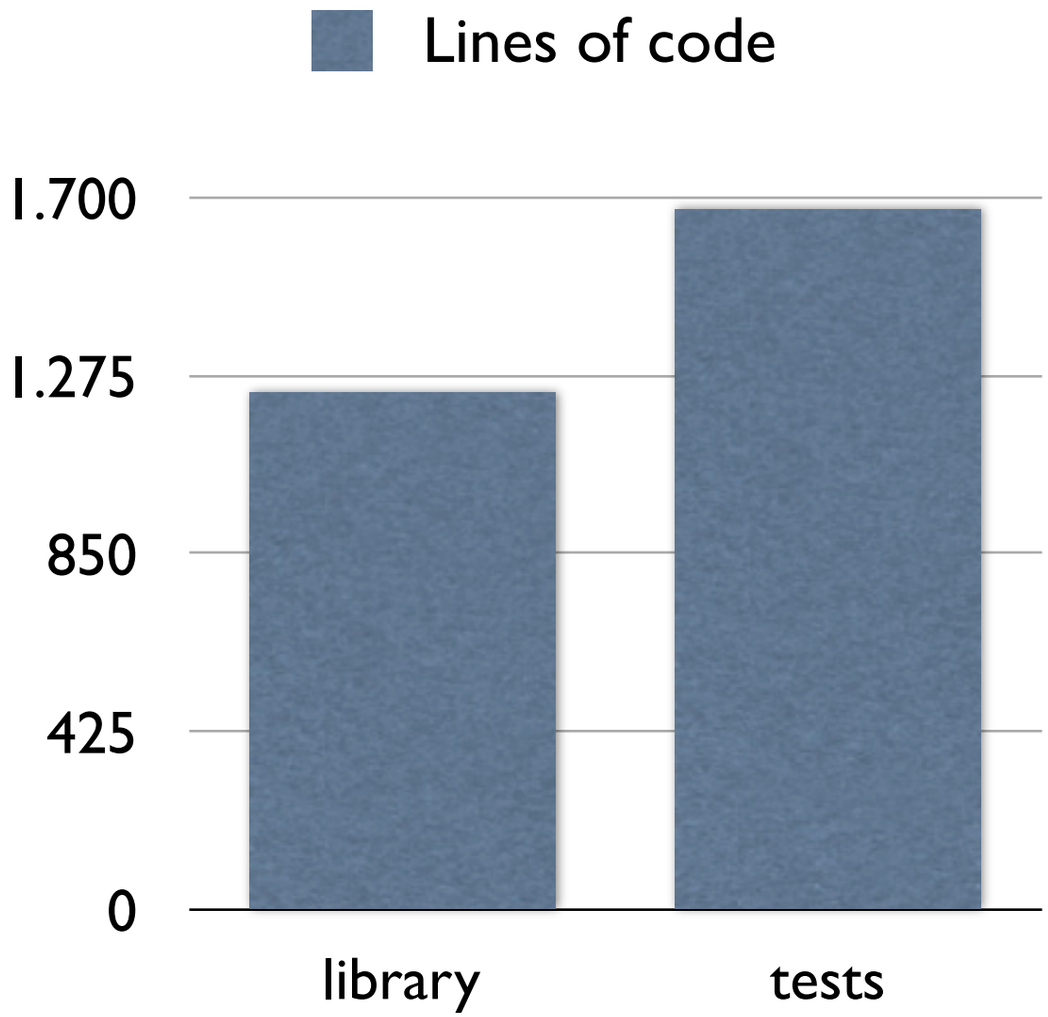
# Transloadit




integration vs. micro tests: 1x : 12.30x

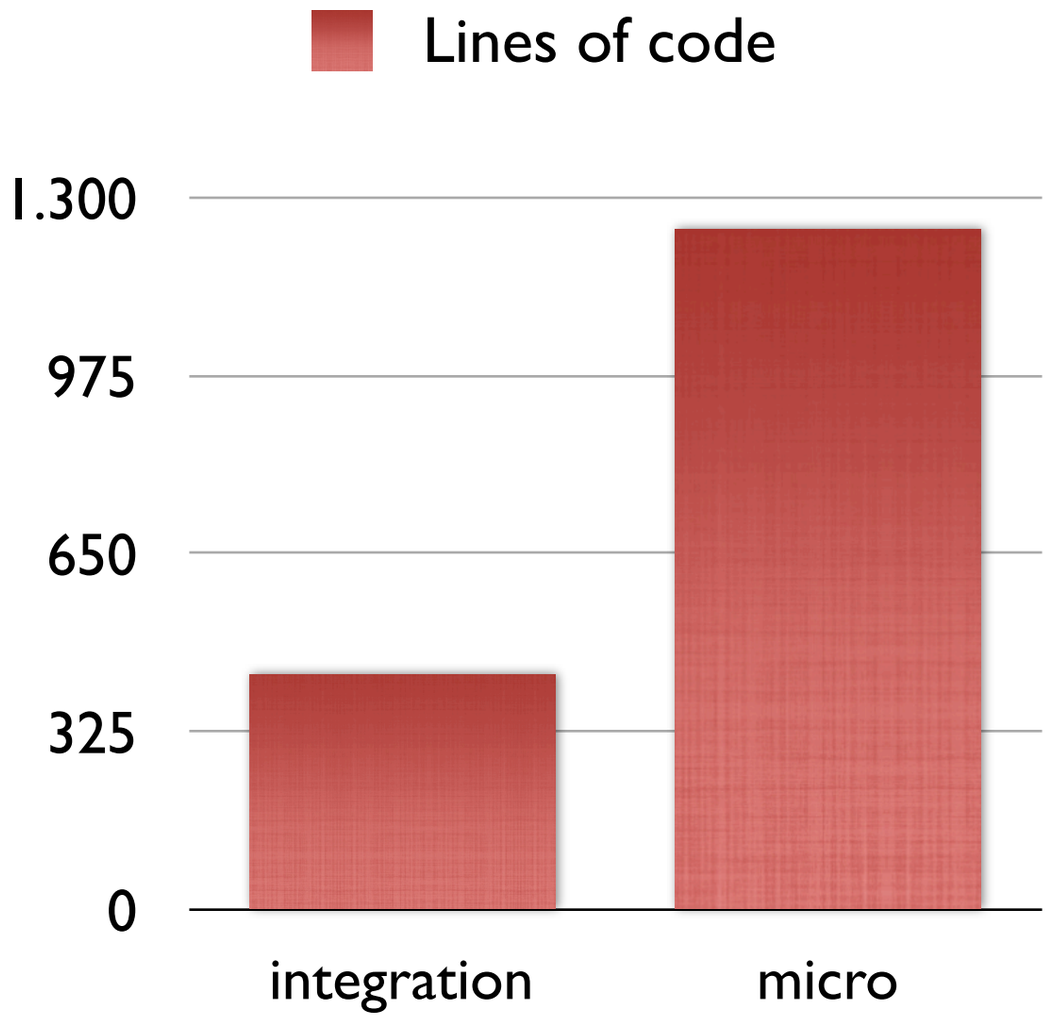
# node-mysql

# node-mysql



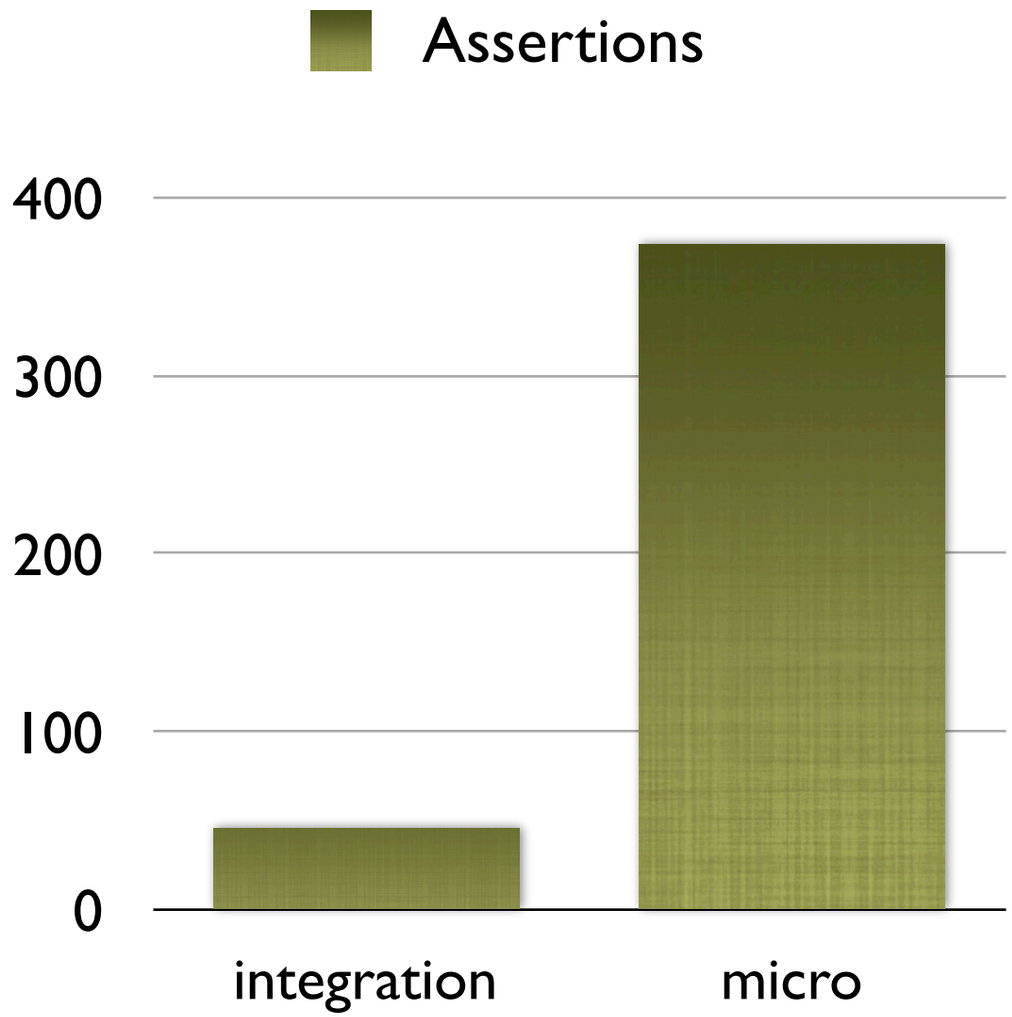

library vs test code: 1x : 1.35x

# node-mysql




integration vs micro tests: 1x : 2.89x

# node-mysql




integration vs micro tests: 1x : 8.15x

Why do you have brakes in a car?
Because then you can drive faster.

*Kevlin Henney*

# tl;dr

- Don't use integration tests to show the basic correctness of your software.

- Write more microtests.

# Questions?

# Questions?

# Thank You

# Thank You