SPEEDING UP YOUR CI

SELF INTRO

- JOSHUA KAPLAN
- MINNE @ GMO PEPABO
- DANGER-SWIFT CONTRIBUTER
- LIKES 🤖

WHAT DO WE USE CI FOR?

1. AUTOMATED TESTING
2. STATIC ANALYSIS (LINTING ETC)
3. BETA DEPLOYMENT
4. RELEASE BUILD DEPLOYMENT

WHAT WERE BUILD TIMES LIKE?

- 1ST BUILD: 55 MINUTES
- 2ND AND LATER: 35 MINUTES

WHAT DID THEY BECOME?

> 14-16 MINUTES

HOW?

CACHING FRAMEWORKS

- MOVING TO CARTHAGE AND USING ROME
  - BIGGEST IMPACT: 55 TO 33 MINUTES

WHAT IS ROME?

UPGRADING PLAN

bitrise.io

bitrise WE'RE HIRING

Why choose Elite plan?

Standard

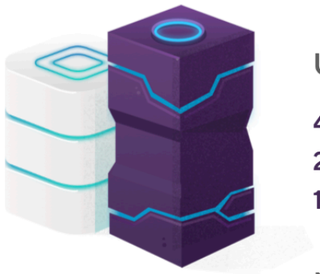
BUILD MACHINES

Ubuntu Virtual Machines

2 vCPU
2.6 GHz Intel Xeon E5
7.5 GB RAM

MacOS Virtual Machines

2 vCPU
Intel® Xeon® CPU X5570 @ 2.93GHz
4 GB RAM



Elite

BUILD MACHINES

Ubuntu Virtual Machines

4 vCPU
2.6 GHz Intel Xeon E5
15 GB RAM

MacOS Virtual Machines

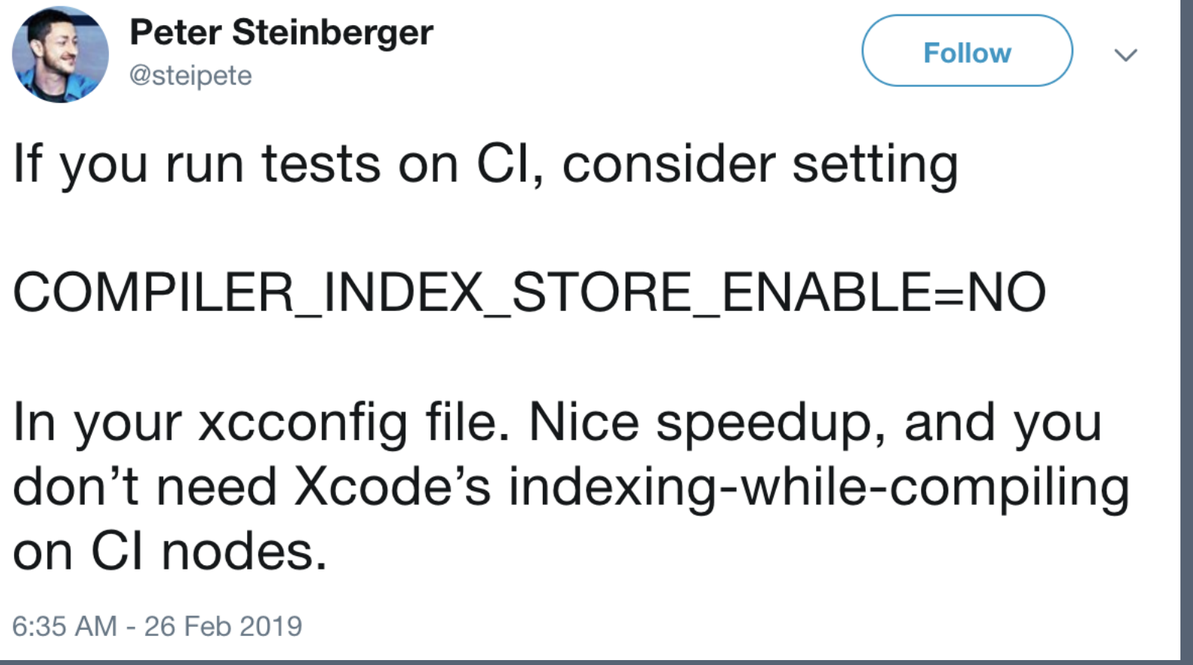
4 vCPU
Intel® Xeon® CPU E5-2697 v2 @ 2.70GHz
8 GB RAM

- NOT MUCH TO EXPLAIN HERE
  - 33 TO 20 MINUTES

ONLY GETTING CODE COVERAGE WHEN TESTS CHANGE

> 2 MINUTES LESS

REMOVE POINTLESS INDEXING


Peter Steinberger
@steipete
Follow
If you run tests on CI, consider setting
COMPILER_INDEX_STORE_ENABLE=NO
In your xcconfig file. Nice speedup, and you don’t need Xcode’s indexing-while-compiling on CI nodes.
6:35 AM - 26 Feb 2019

> SAVES 1 OR 2 MINUTES

```
COMPILER_INDEX_STORE_ENABLE=NO
```

OTHER TECHNIQUES I LOOK FORWARD TO

- PARALLELIZATION
- SEPARATE MODULES

CONCLUSION

MESS AROUND