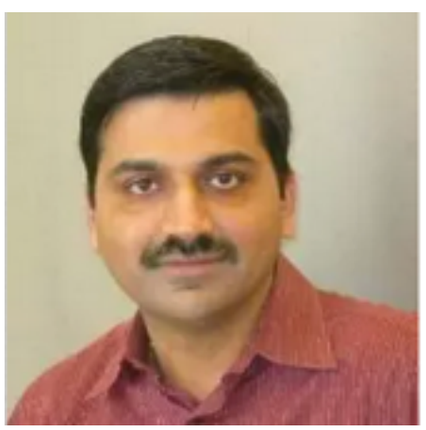
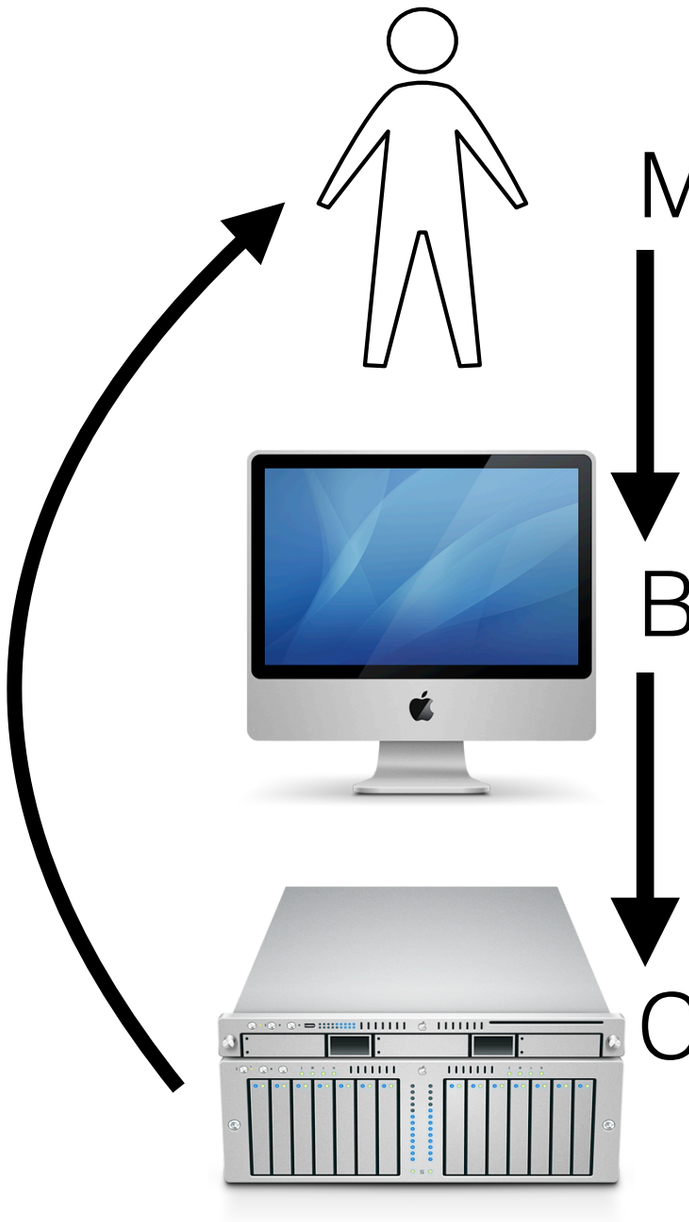
Efficient Dependency Detection for Safe Java Test Acceleration

Jonathan Bell, Gail Kaiser, Eric Melski and Mohan Dattatreya
Columbia University & Electric Cloud, Inc



Simplified Software Lifecycle



Make changes to code

Build & test

Commit

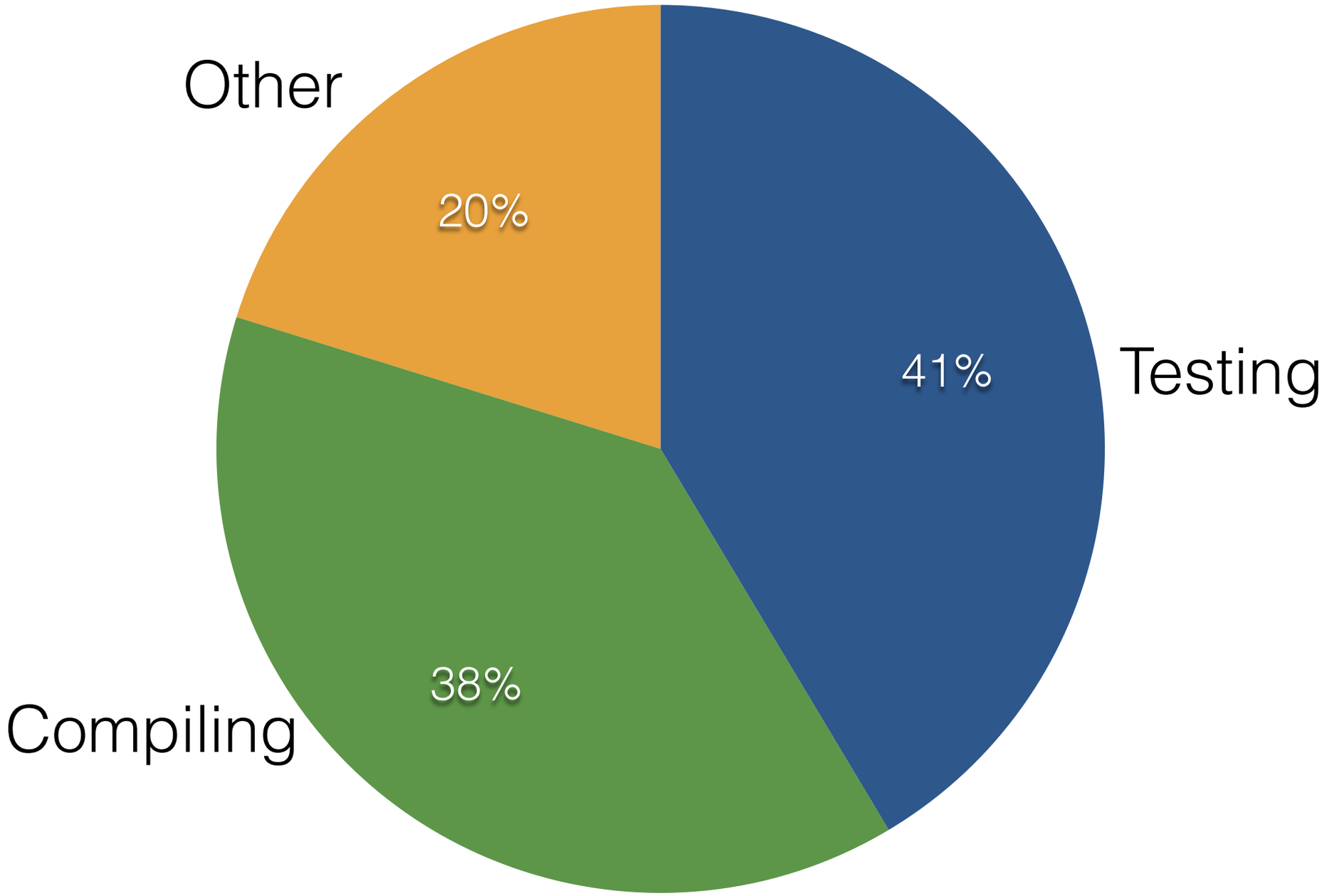
How long is too long of a build?

1 day? **6 hours?** **10 minutes?**

Simplified Software Lifecycle

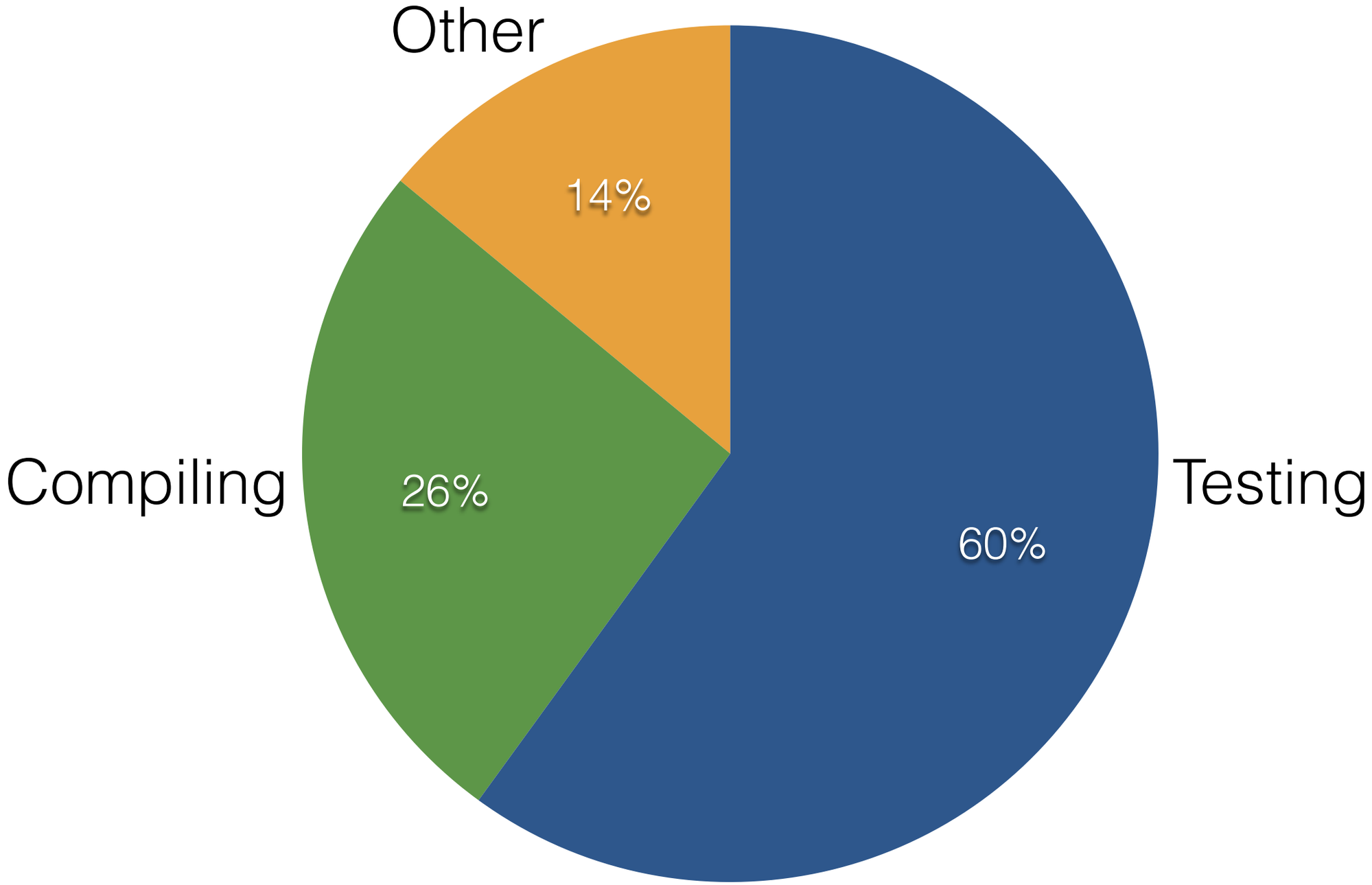
- Compile sources
- Generate documentation
- Run tests
- Package

Testing Dominates Build Times



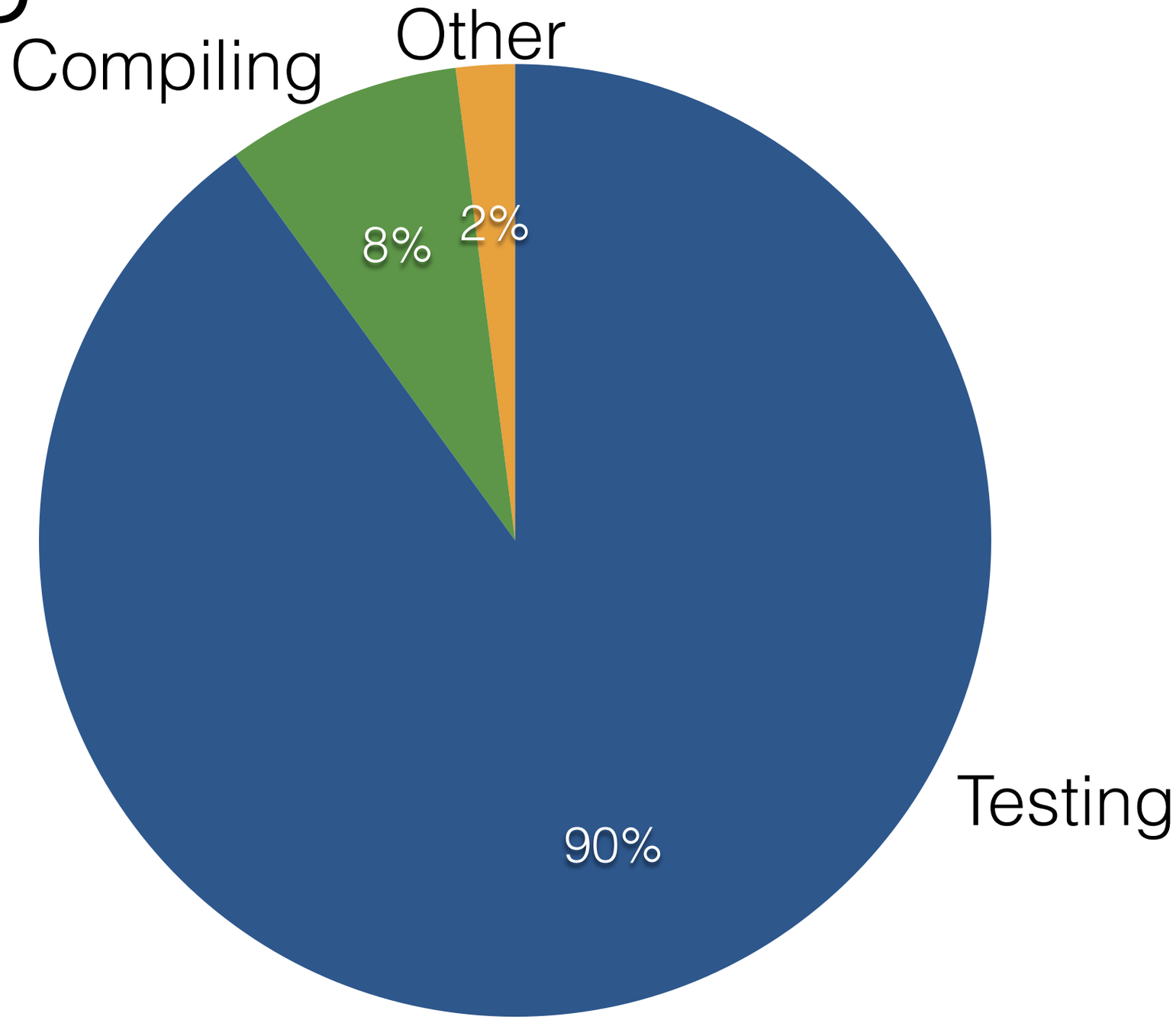
351 projects from GitHub

Testing Dominates Build Times



Projects taking > 10 minutes to build (69)

Testing Dominates Build Times

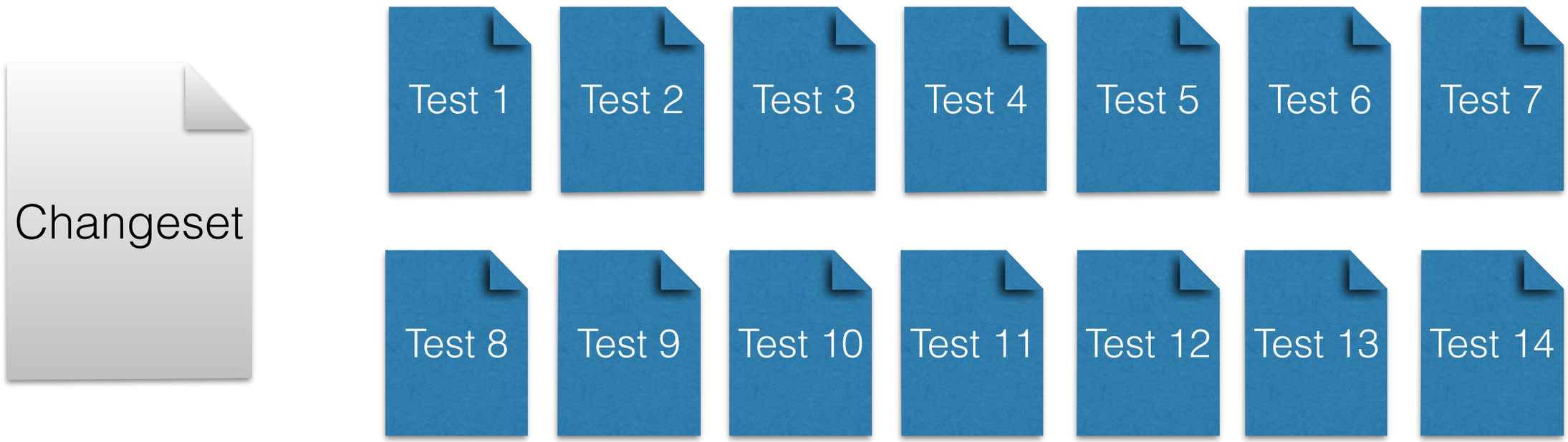


Projects taking > 1 hour to build (8)

Faster tests = Faster builds

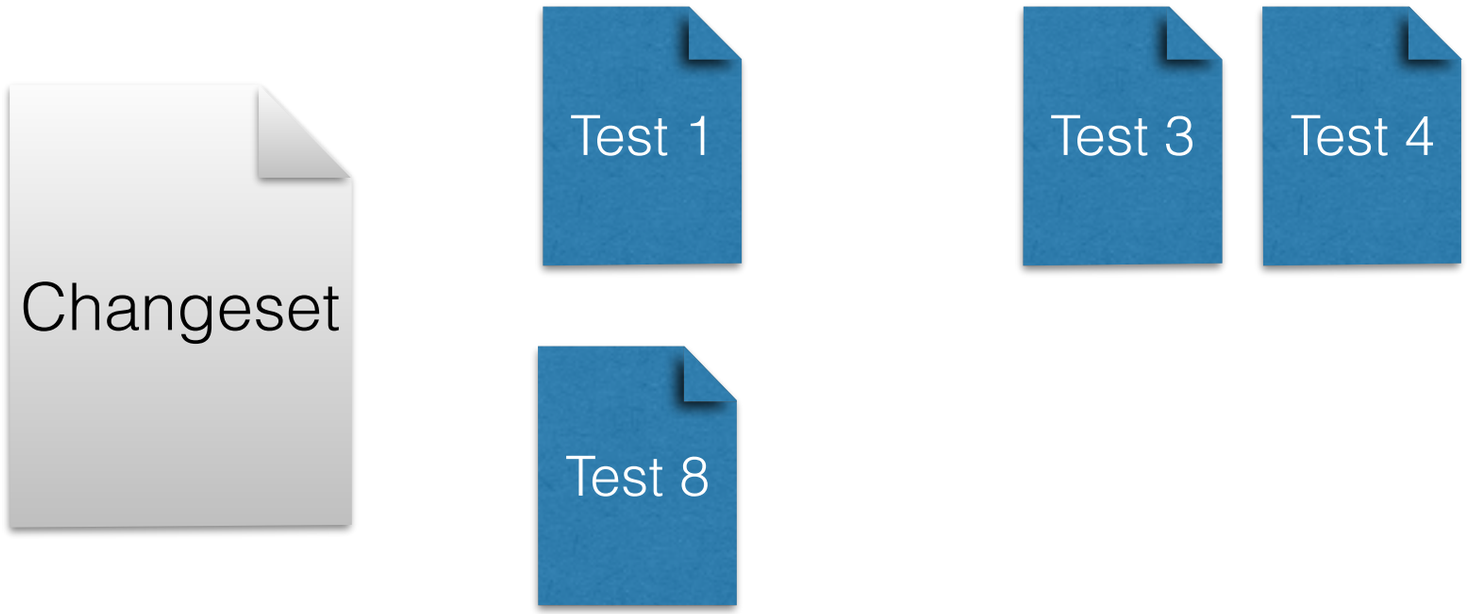
Test acceleration is well studied

Regression Test Selection



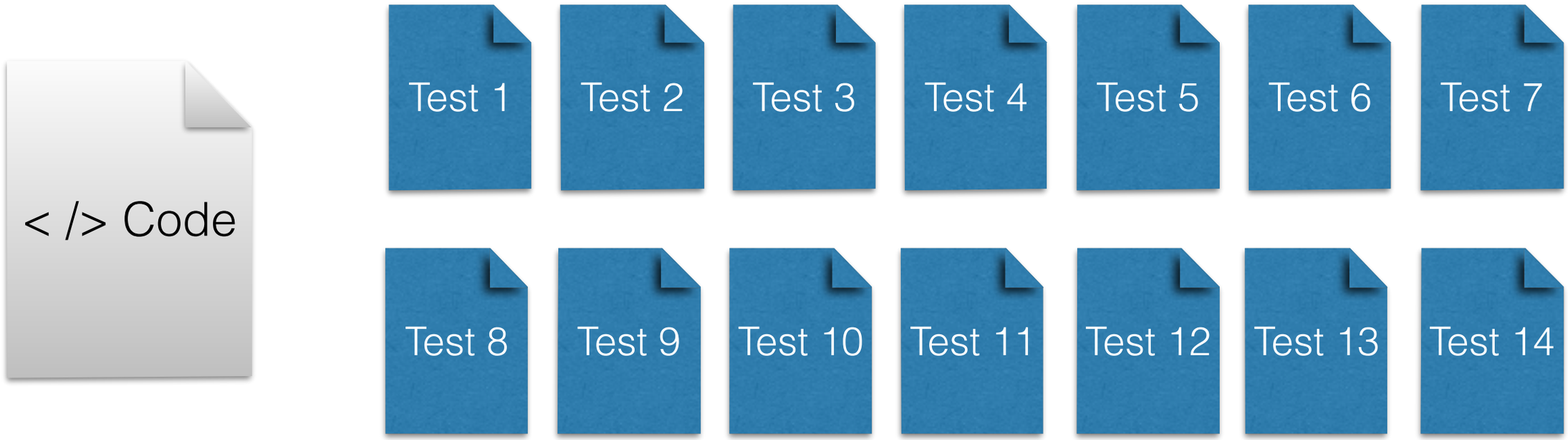
Gligoric et al. [ISSTA ’15], Orso et al. [FSE ’04], Harrold et al. [OOPSLA ’01]

Regression Test Selection



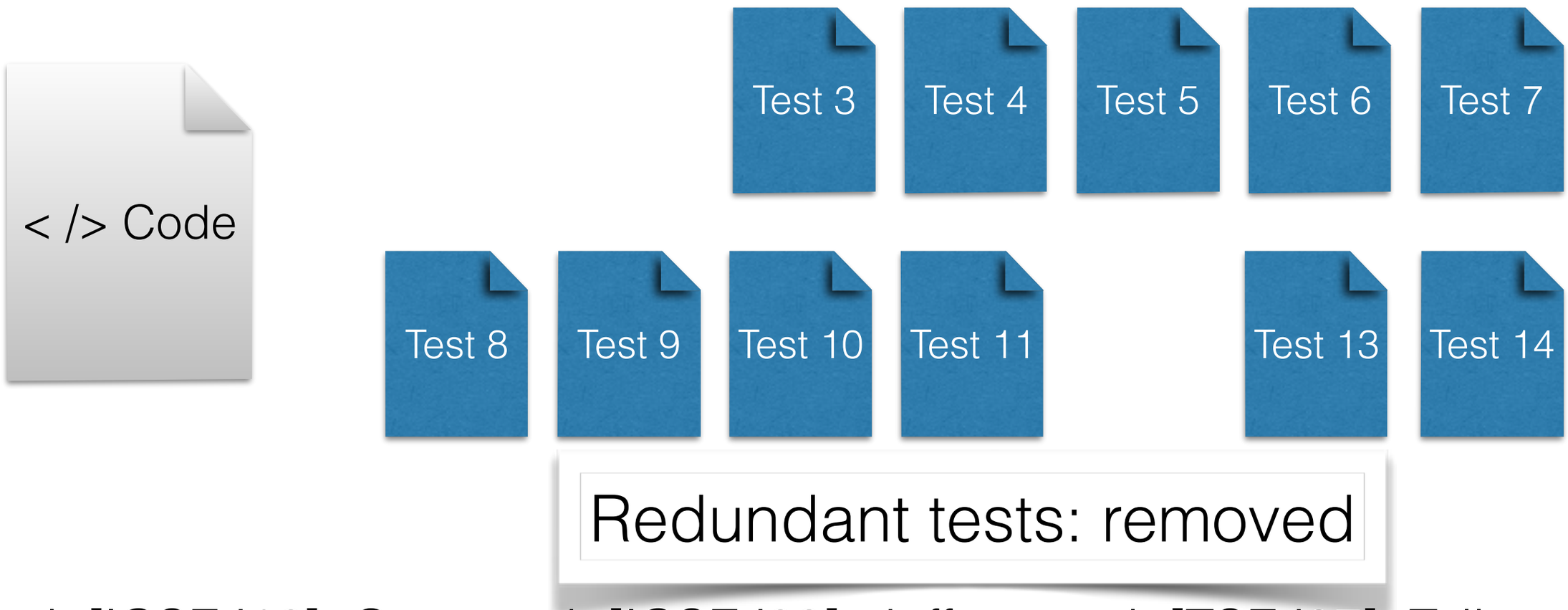
Gligoric et al. [ISSTA ’15], Orso et al. [FSE ’04], Harrold et al. [OOPSLA ’01]

Test Suite Minimization



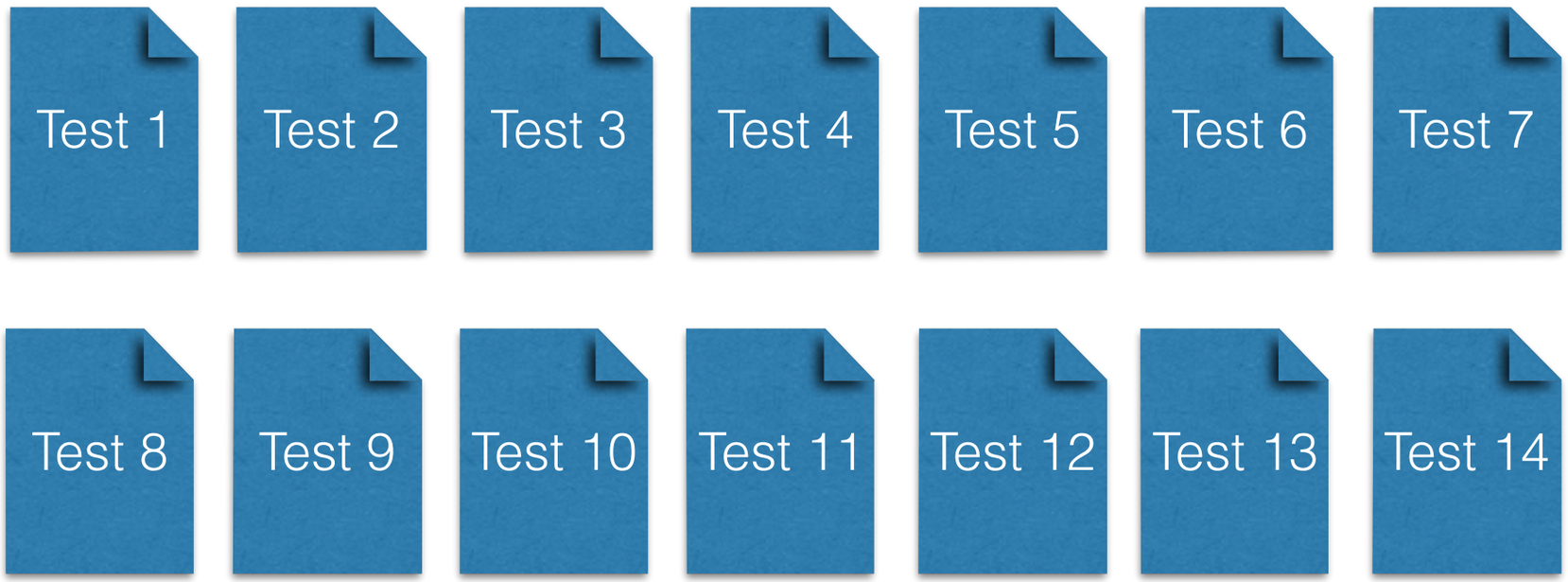
Hao et al. [ICSE '12]; Orso et al. [ICSE '09]; Jeffrey et al. [TSE '07]; Tallam et al. [PASTE '05]; Jones et al. [TOSEM '03]; Harrold et al. [TOSEM '93]; Chen et al. [IST '98]; Wong et al. [ICSE '95] and more

Test Suite Minimization

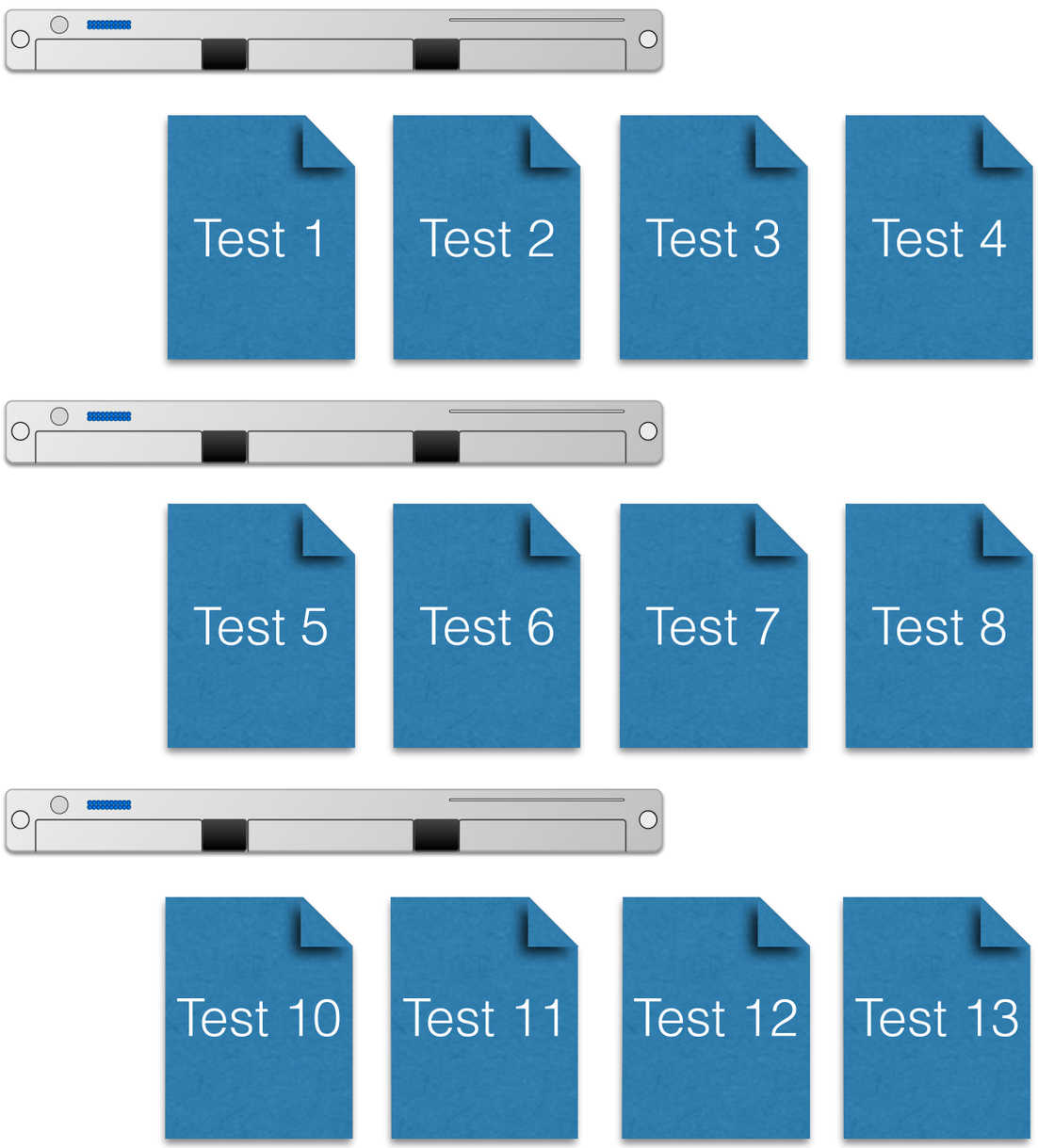


Hao et al. [ICSE ’12]; Orso et al. [ICSE ’09]; Jeffrey et al. [TSE ’07]; Tallam et al. [PASTE ’05]; Jones et al. [TOSEM ’03]; Harrold et al. [TOSEM ’93]; Chen et al. [IST ’98]; Wong et al. [ICSE ’95] and more

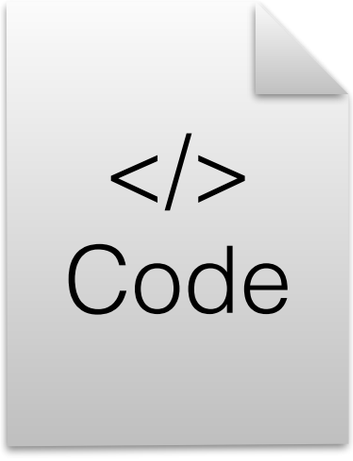
Test Parallelization



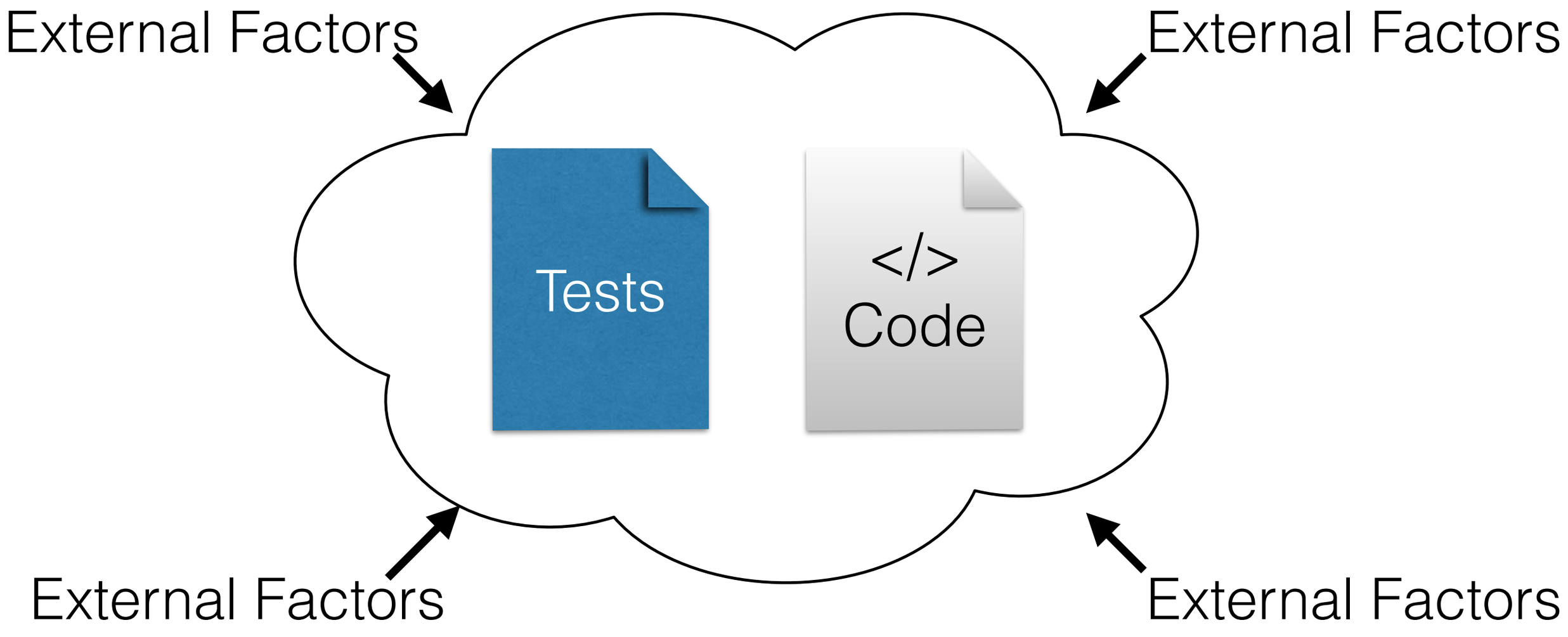
Test Parallelization



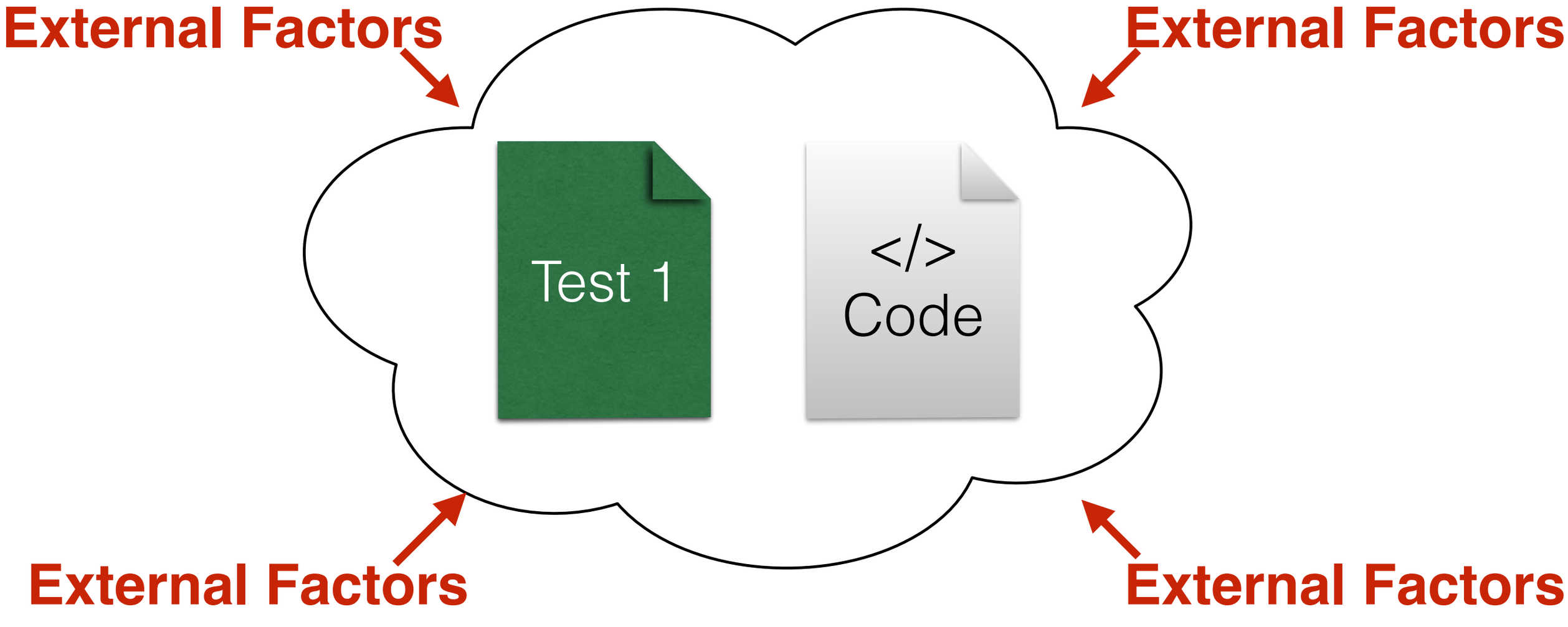
Controlled Regression Testing Assumption



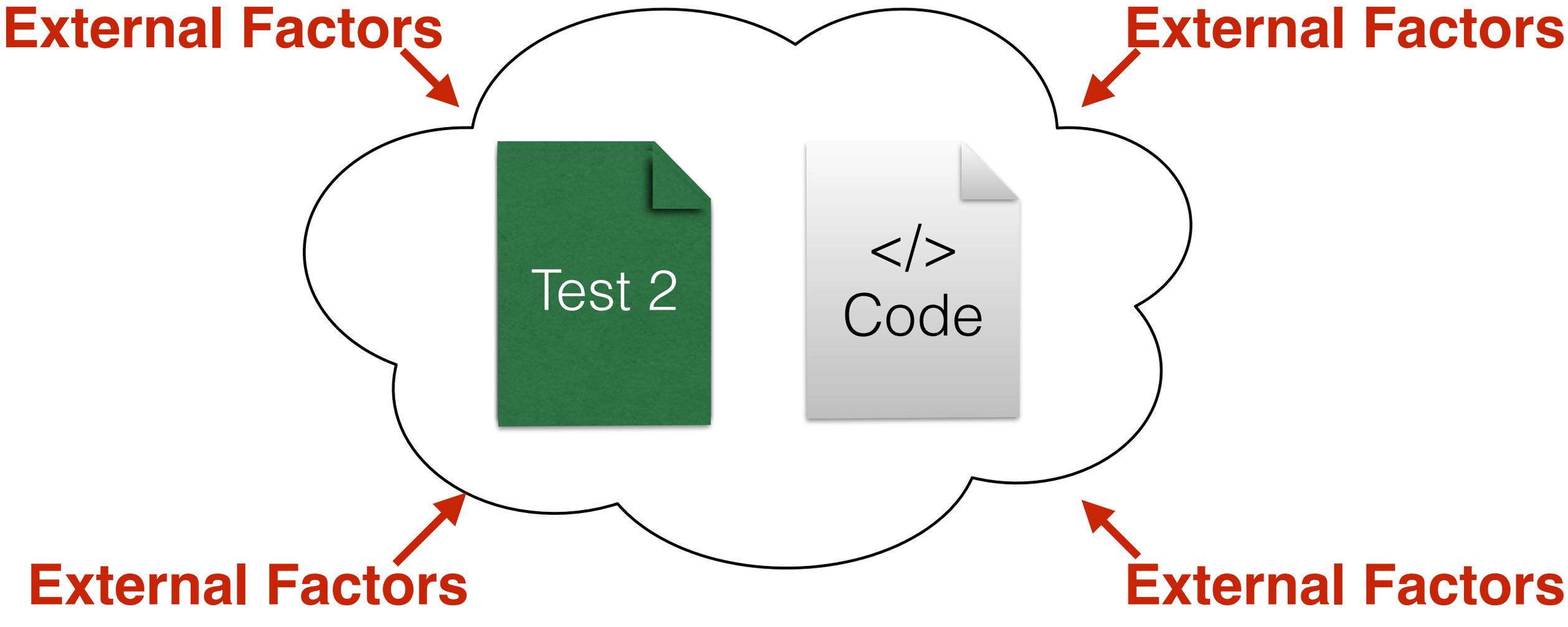
Controlled Regression Testing Assumption



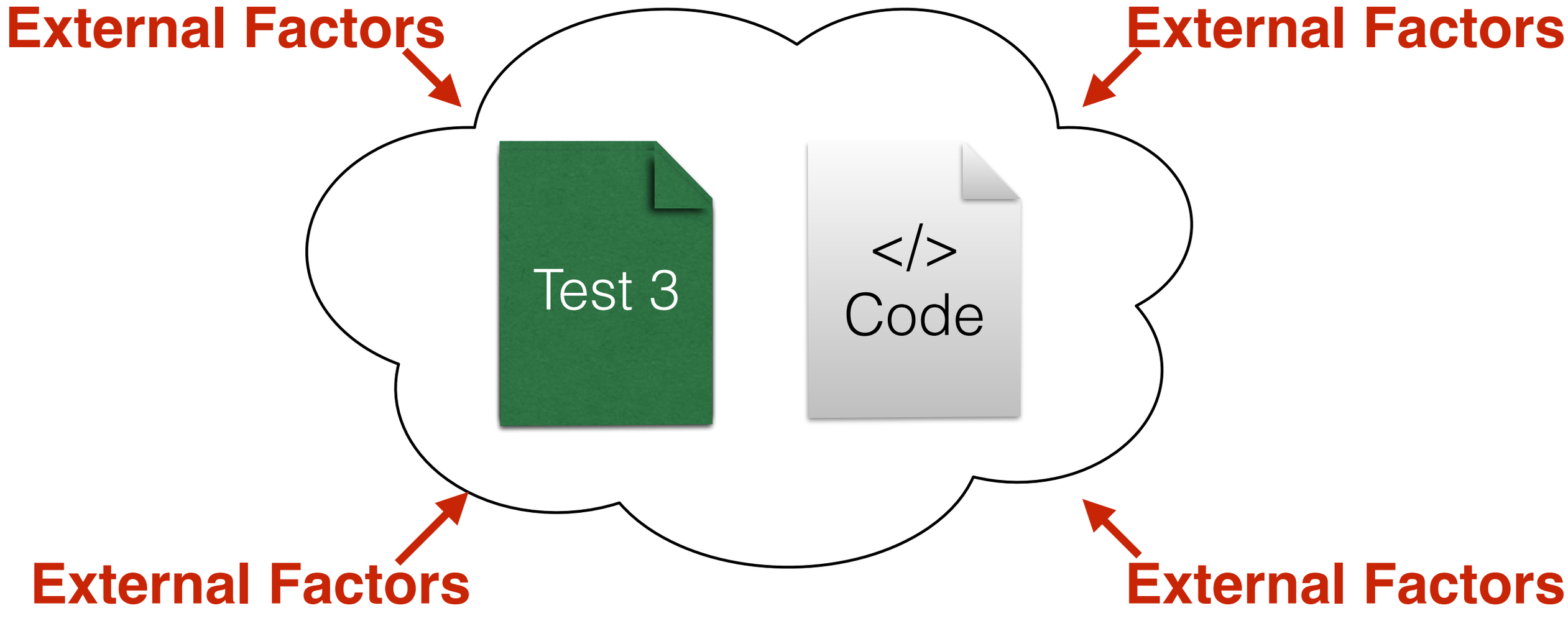
Controlled Regression Testing Assumption



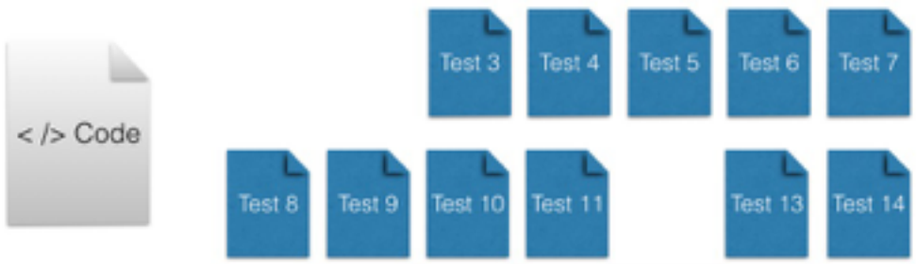
Controlled Regression Testing Assumption



Controlled Regression Testing Assumption



Test Suite Minimization



Redundant tests: removed

Hao et al. [ICSE '12]; Orso et al. [ICSE '09]; Jeffrey et al. [TSE '07]; Tallam et al. [PASTE '05]; Jones et al. [TOSEM '03]; Harrold et al. [TOSEM '93]; Chen et al. [IST '98]; Wong et al. [ICSE '95] and more

Regression Test Selection

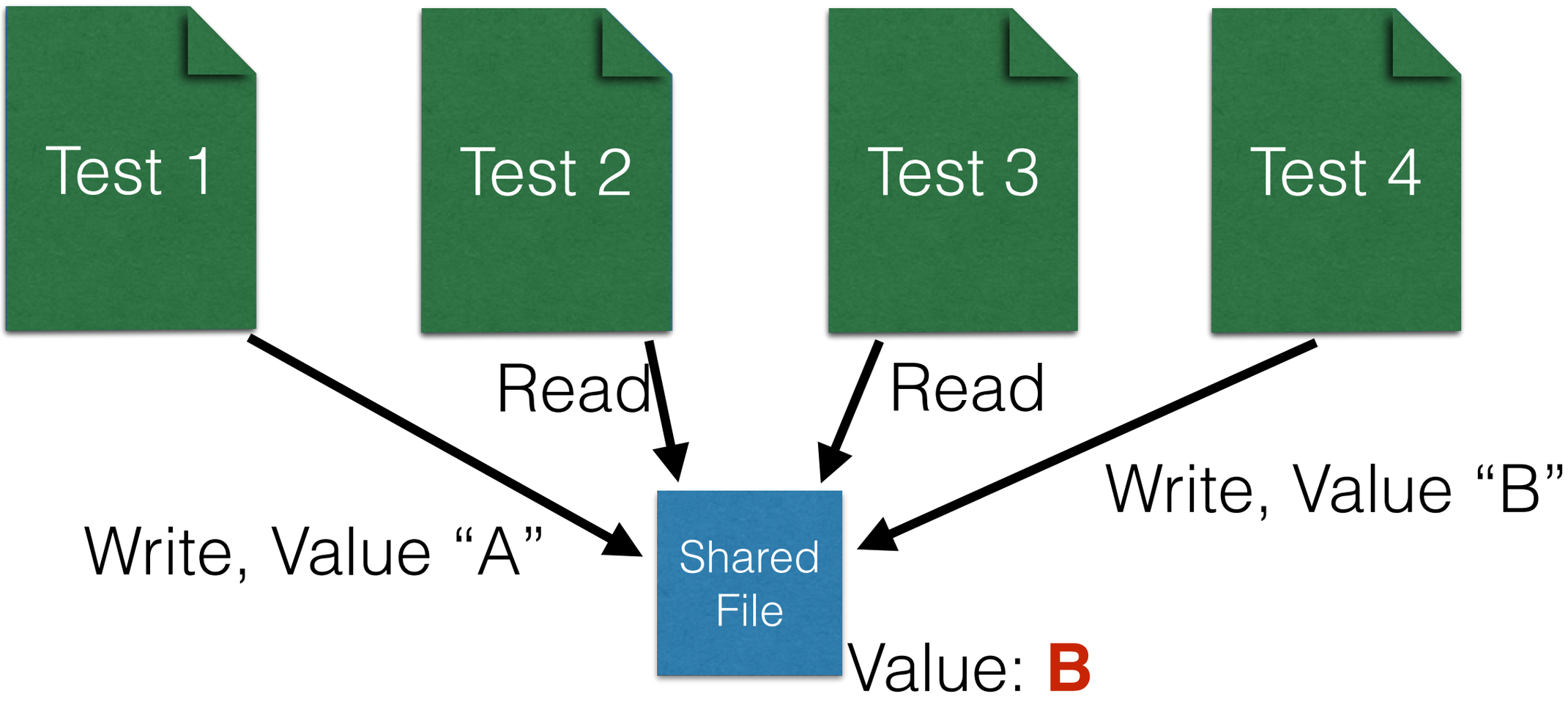
Tests not relevant to changeset: skipped

Gligoric et al. [ISSTA '15], Orso et al. [FSE '04], Harrold et al. [OOPSLA '01]

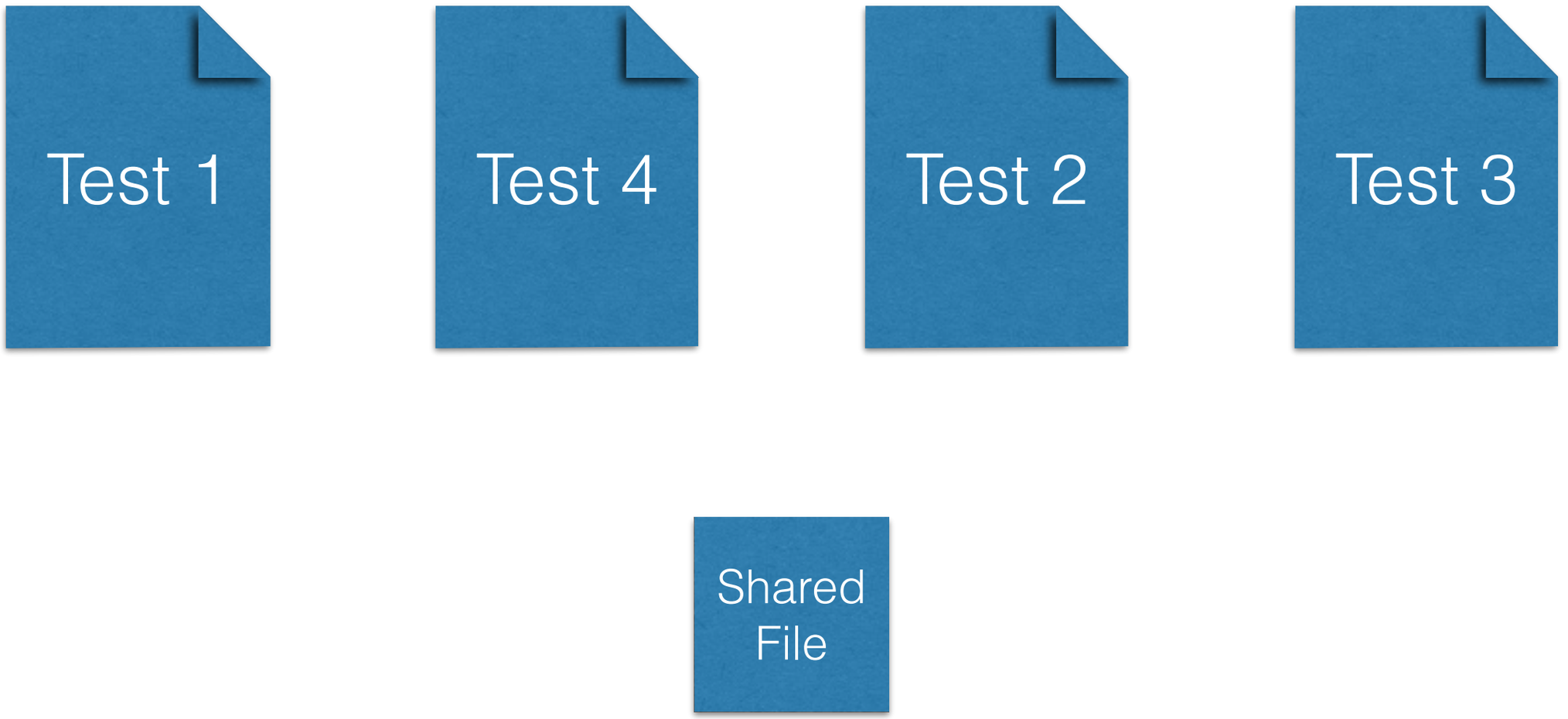
Not sound in practice

Test Parallelization

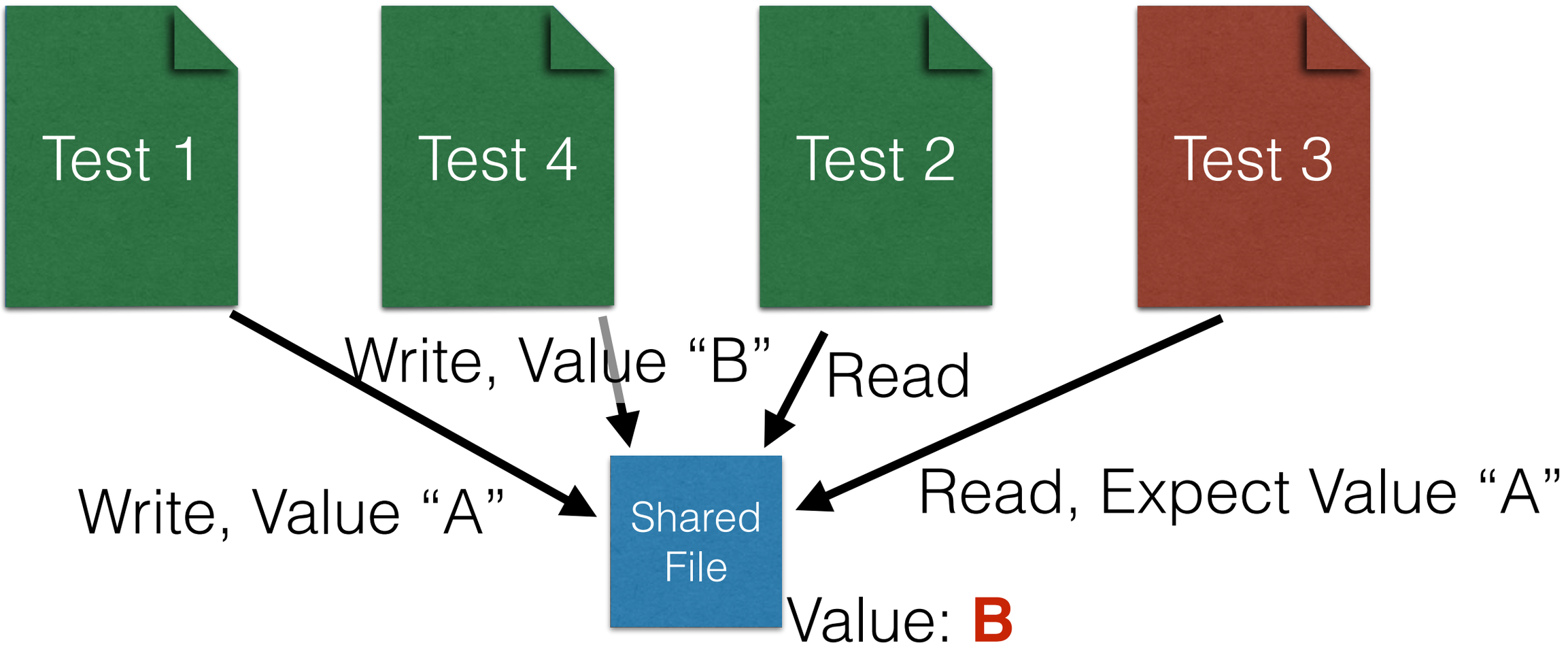
Test Dependencies



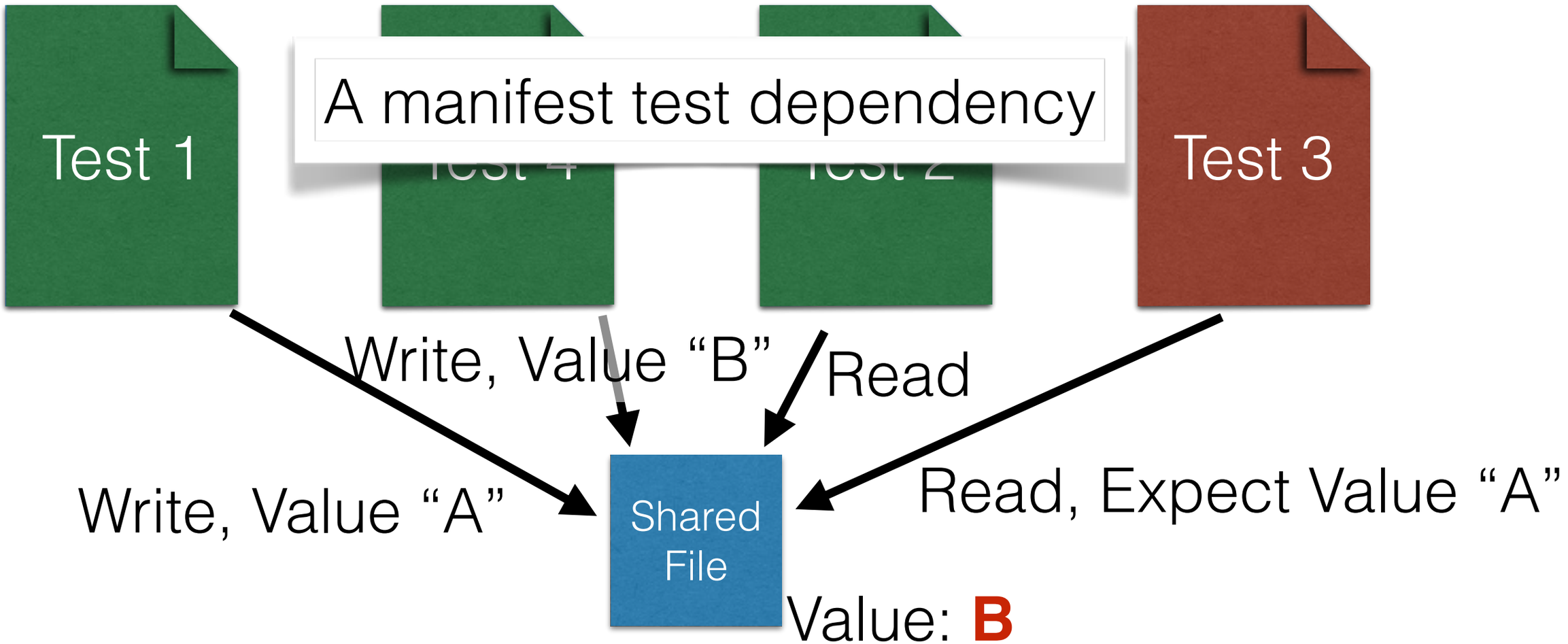
Test Dependencies



Test Dependencies



Test Dependencies



Test Dependencies: A Clear and Present Danger

- Really exist in practice (Zhang et al. found 96, Luo et al. found 14)

- Hard to specify - if we could specify, would be safe to accelerate

- Rarely: isolated (especially not in long building projects)

- Existing technique to detect: combinatorially run tests [Zhang, et al ’14]

Brute Force Dependency Detection

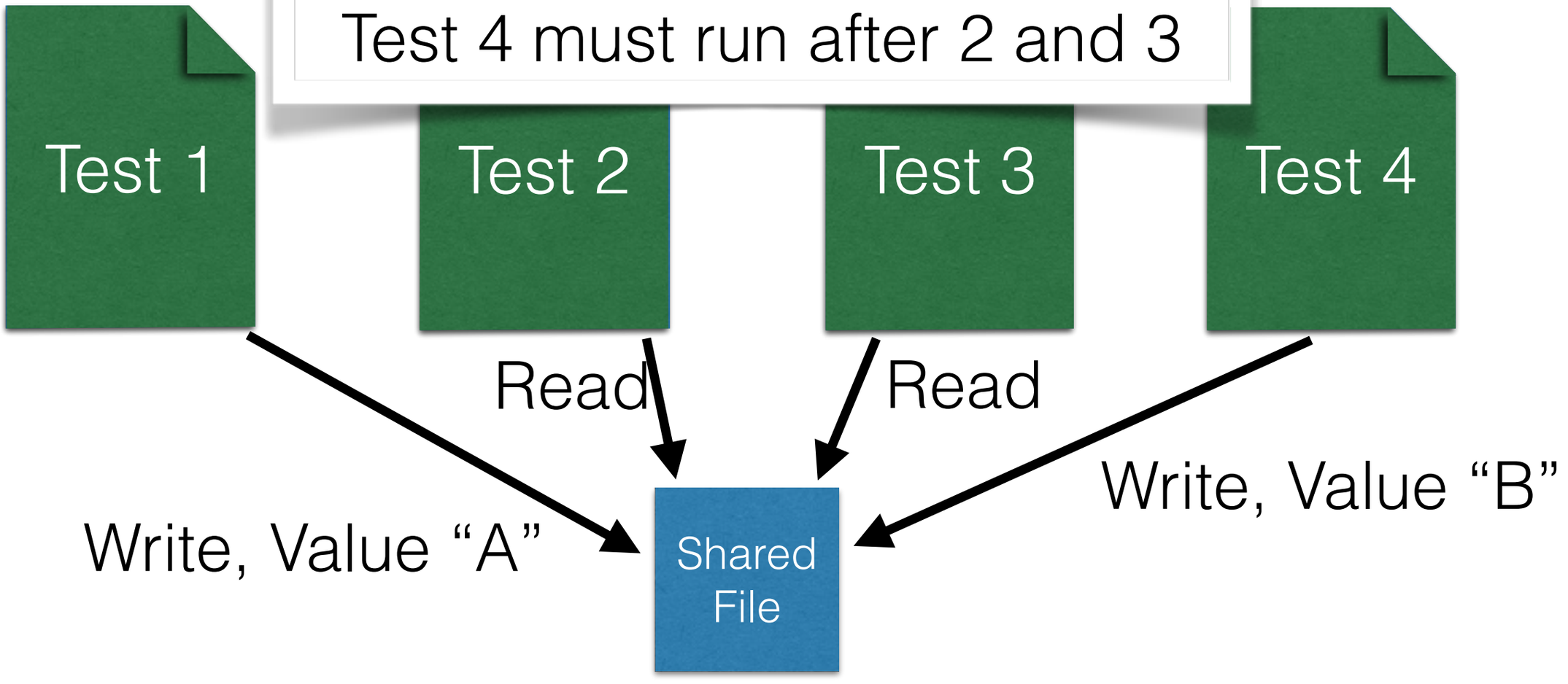
- Looked at feasibility on 10 large open source test suites
- Exhaustive approach: > 10^{300} years to find all dependencies
- Pairwise approach: Average 31,882 executions of the entire test suite to find (incomplete) dependencies
- Problem: How do we safely accelerate test suites in the presence of unknown dependencies?

Manifest Test Dependencies

- Definition: a data dependence between tests T1, T2 that results in the outcome of T2 changing
  - All manifest dependencies are data dependencies
  - Not all data dependencies are manifest dependencies

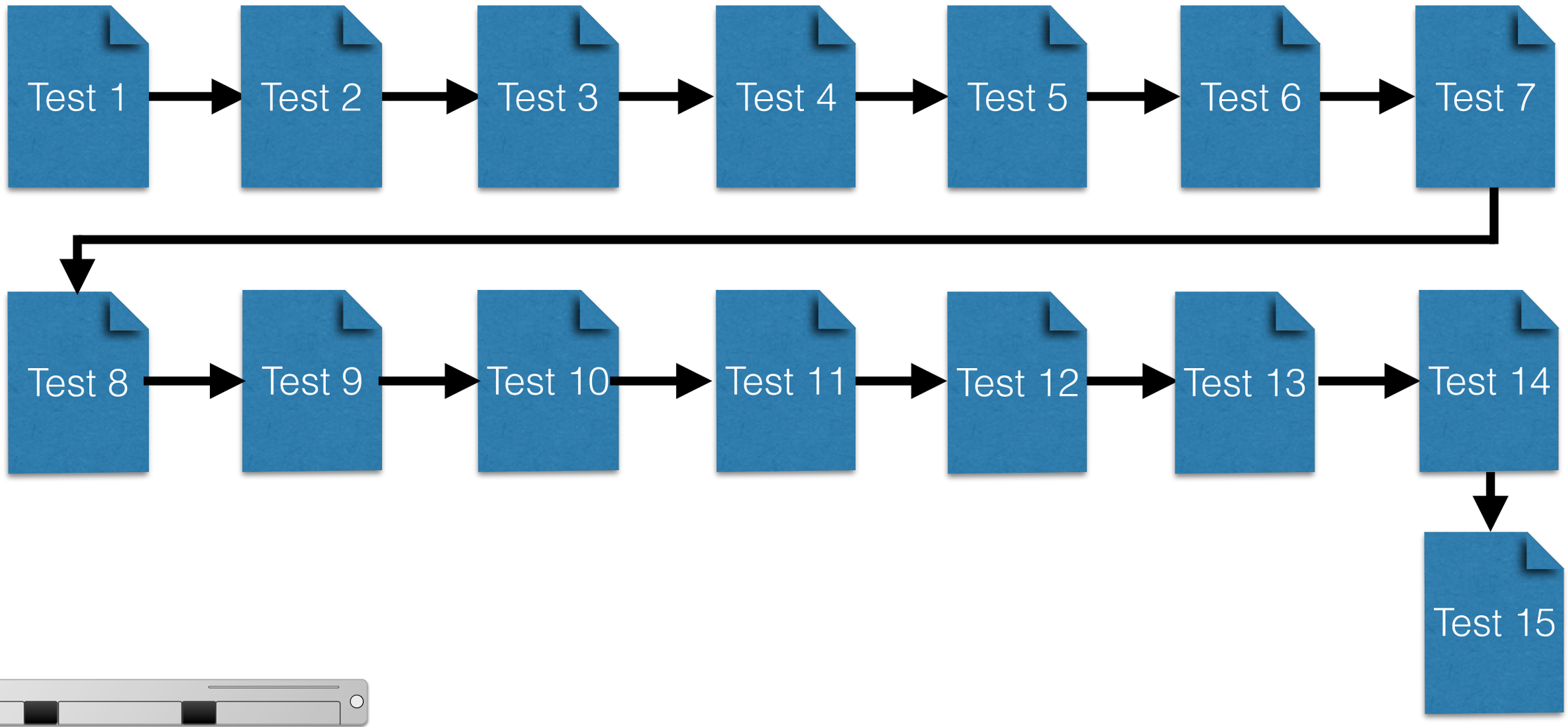
Data Dependencies

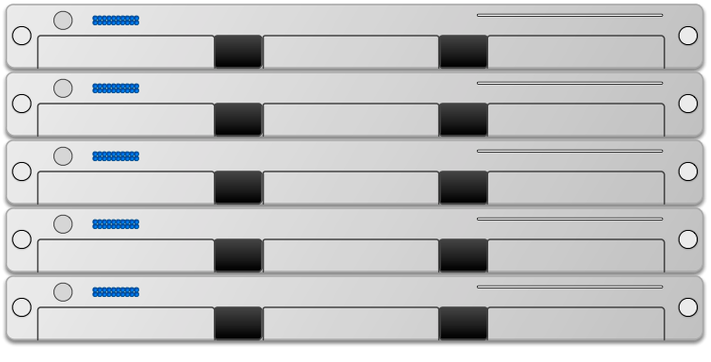
Present Dependencies:
Test 1 must run before 2 and 3
Test 4 must run after 2 and 3



Key Insight: Dependencies don’t need to be **precise**, but must be **sound**

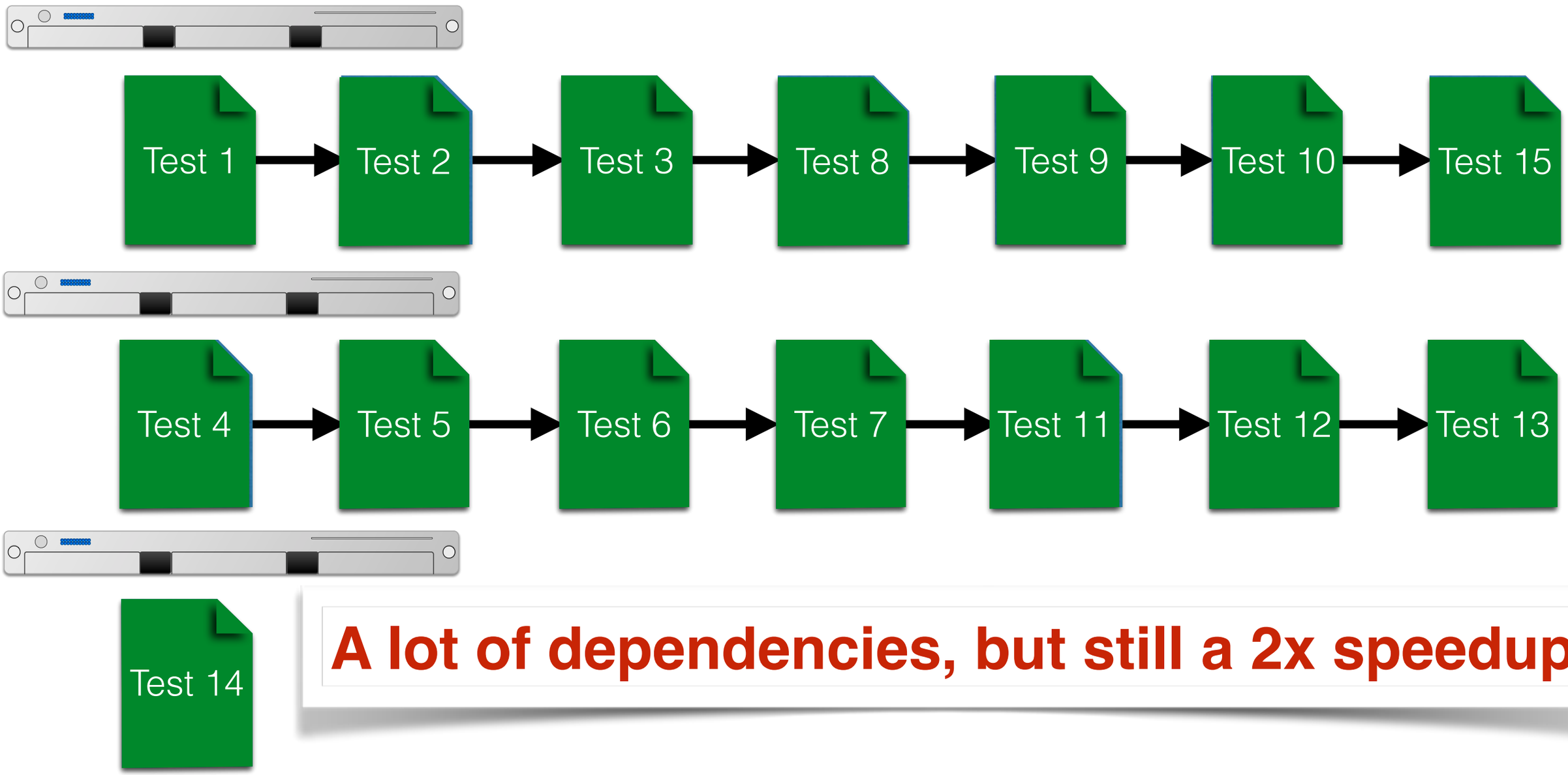
Intuition





Idle extra capacity

Intuition



A lot of dependencies, but still a 2x speedup

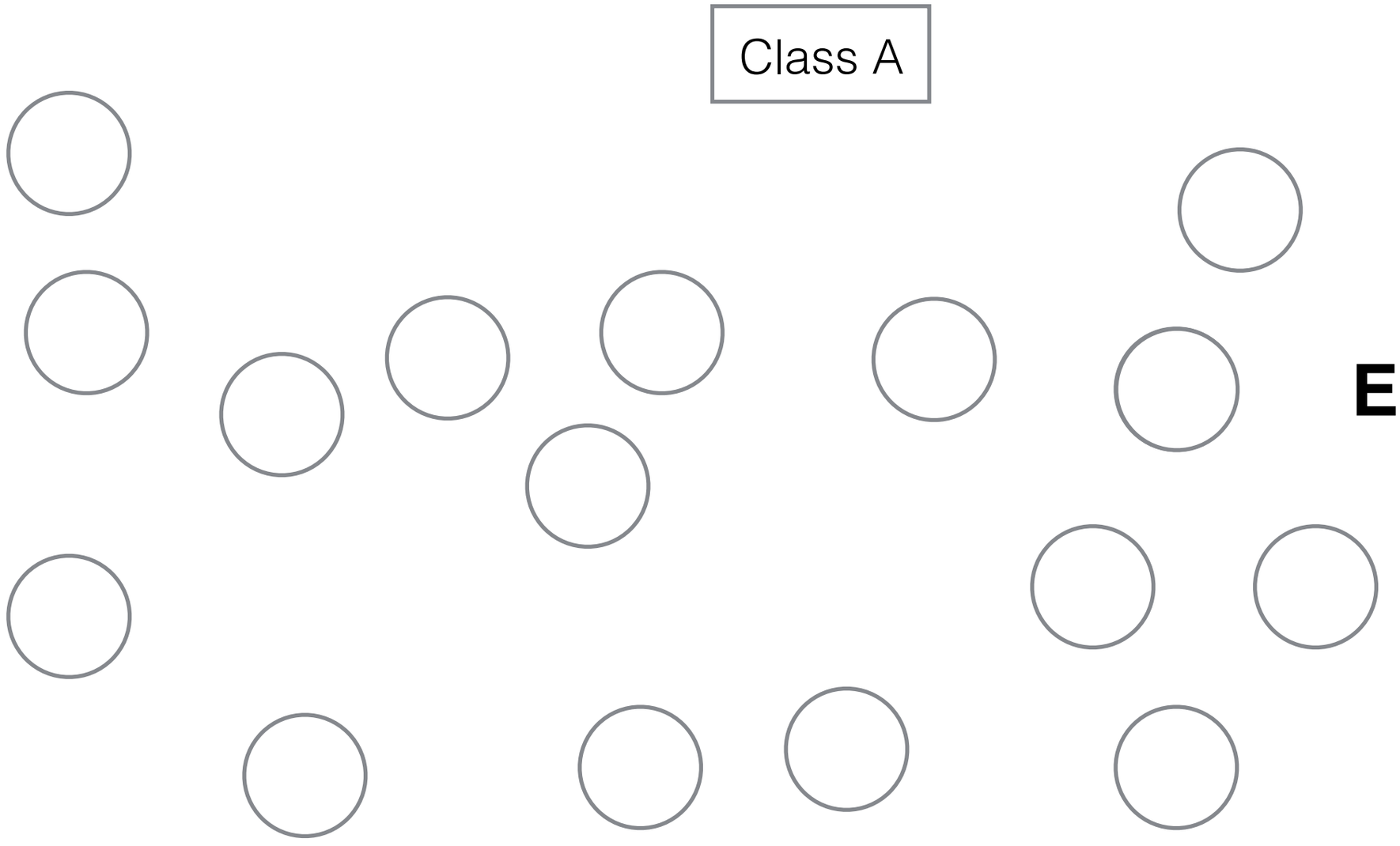
Idle extra capacity

ElectricTest - Detecting Data Dependencies in Java

- Tracks in-memory dependencies (JVMTI plugin)
- Tracks file and network dependencies (IO-Trace agent)
- Implemented entirely within the Oracle or OpenJDK JVM, no specialized drivers, etc required
- Captures stack traces when dependencies occur to support debugging
- Generates dependency trees to enable sound test acceleration

Identifying Heap Dependencies

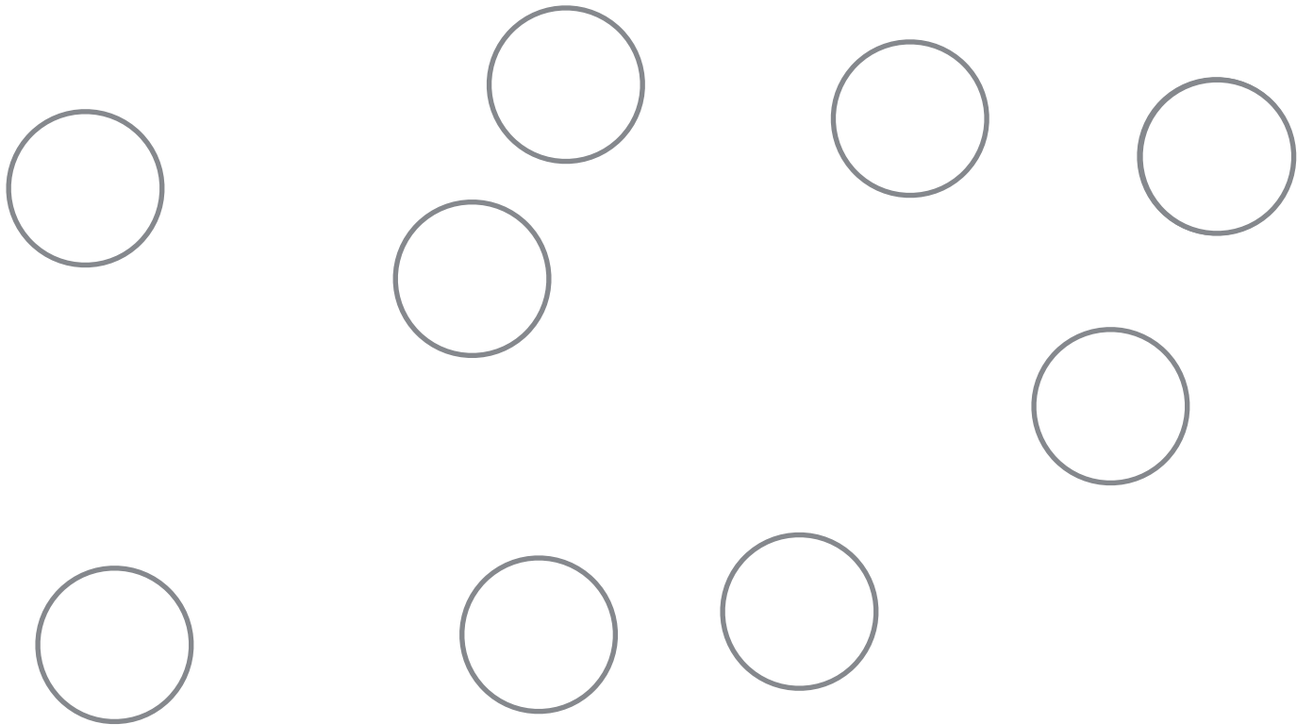
After each test, garbage collect; traverse heap to map objects back to static fields.



Identifying Heap Dependencies

After each test, garbage collect; traverse heap to map objects back to static fields.

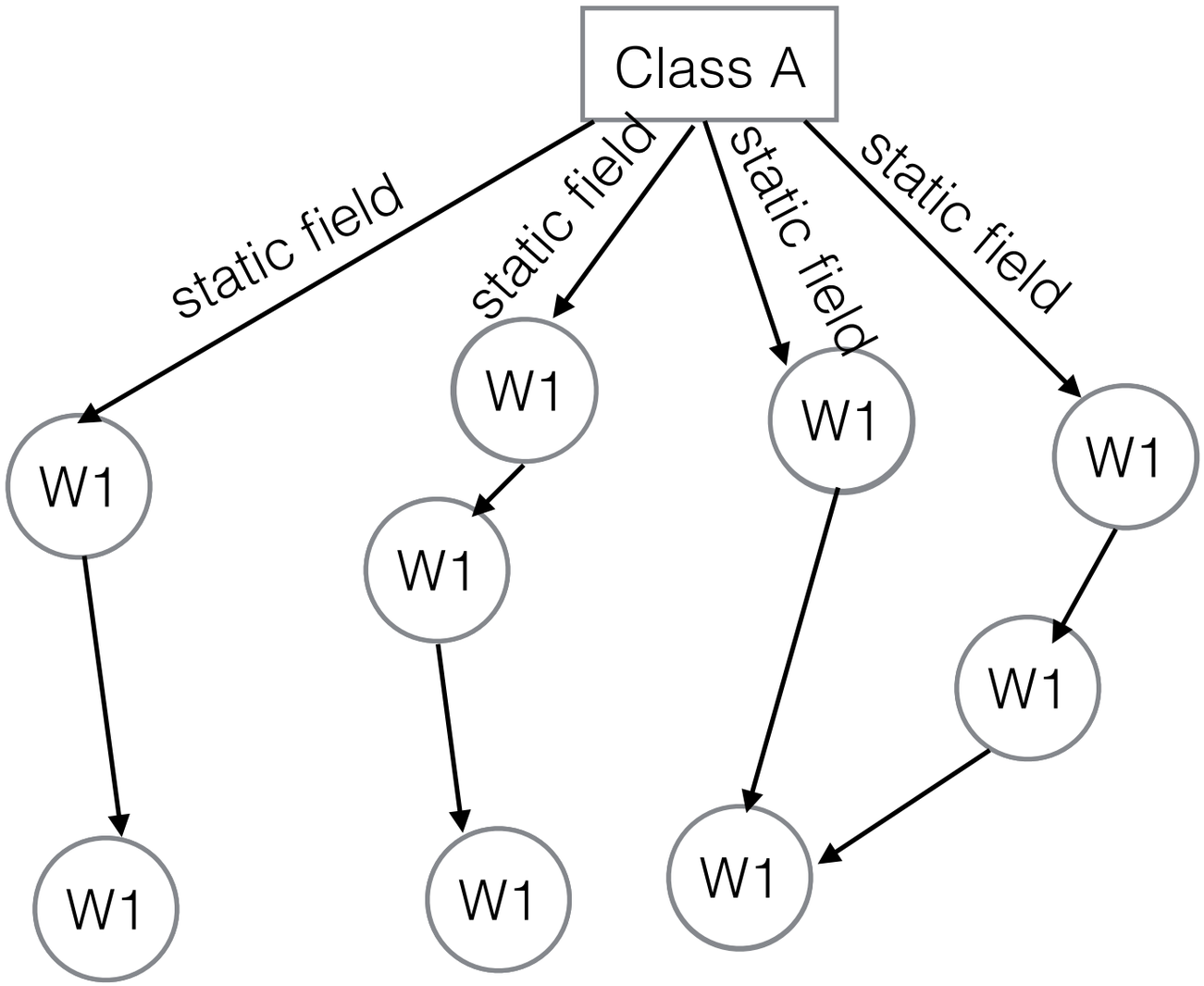
Class A



End of test 1

Identifying Heap Dependencies

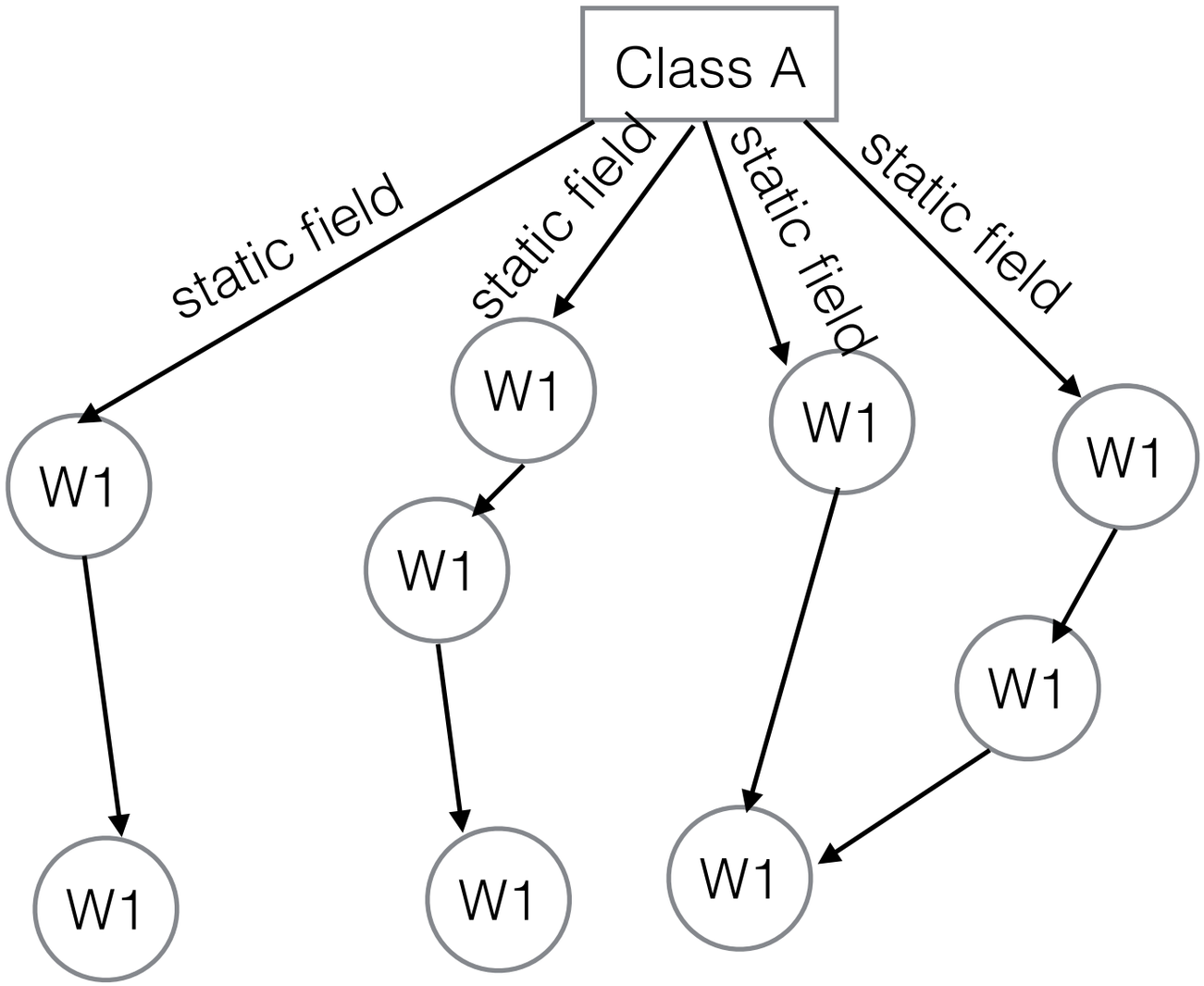
After each test, garbage collect; traverse heap to map objects back to static fields.



End of test 1

Identifying Heap Dependencies

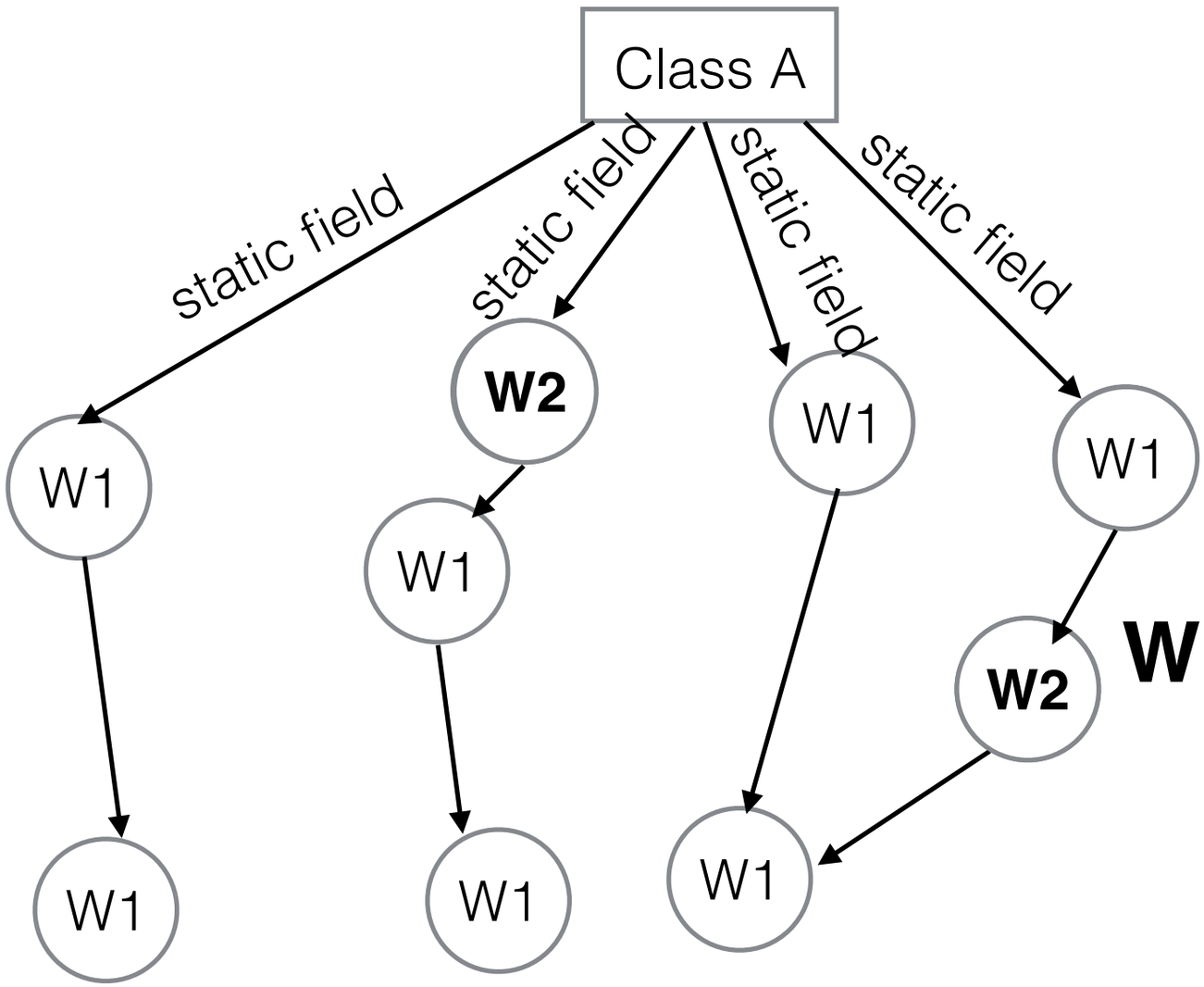
During test execution, monitor accesses to existing objects



During Test 2

Identifying Heap Dependencies

During test execution, monitor accesses to existing objects

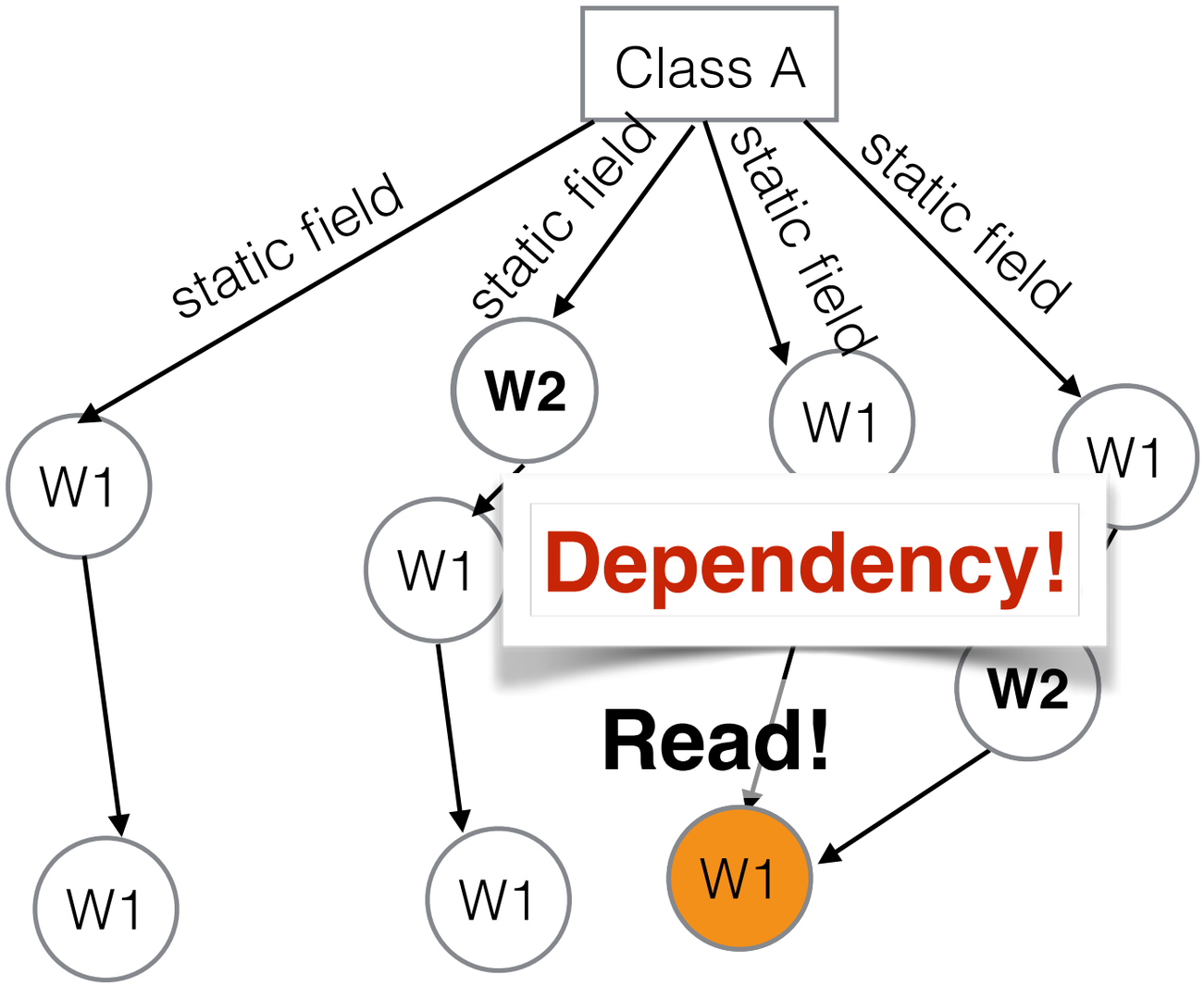


During Test 2

Write!

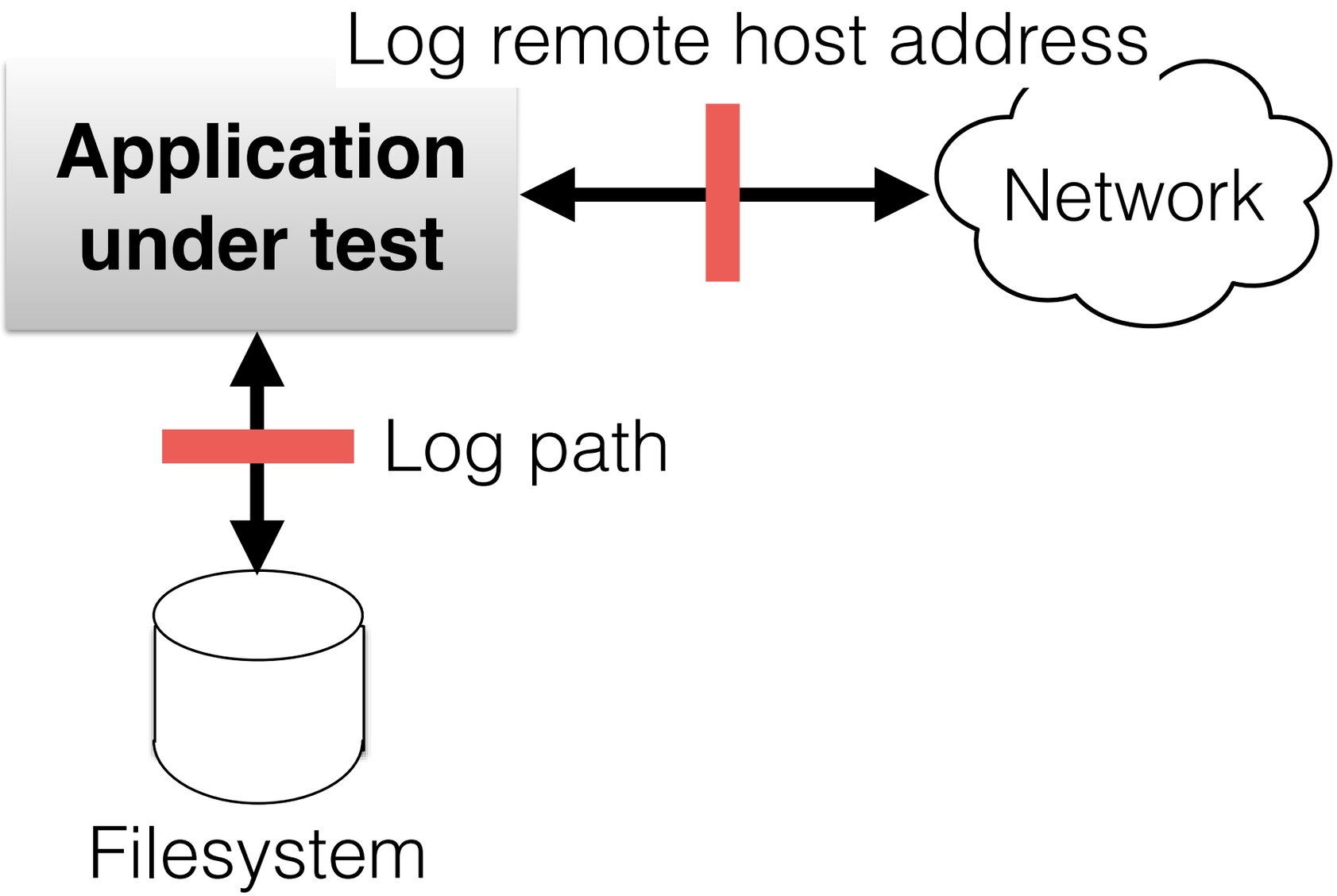
Identifying Heap Dependencies

During test execution, monitor accesses to existing objects



During Test 2

Identifying External Dependencies

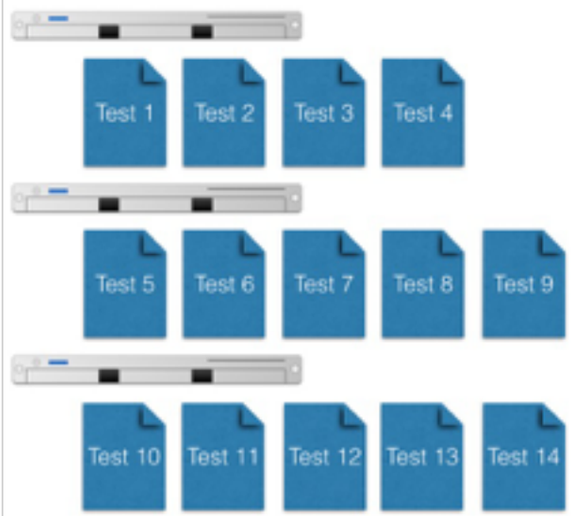


Test Suite Minimization

Regression Test Selection

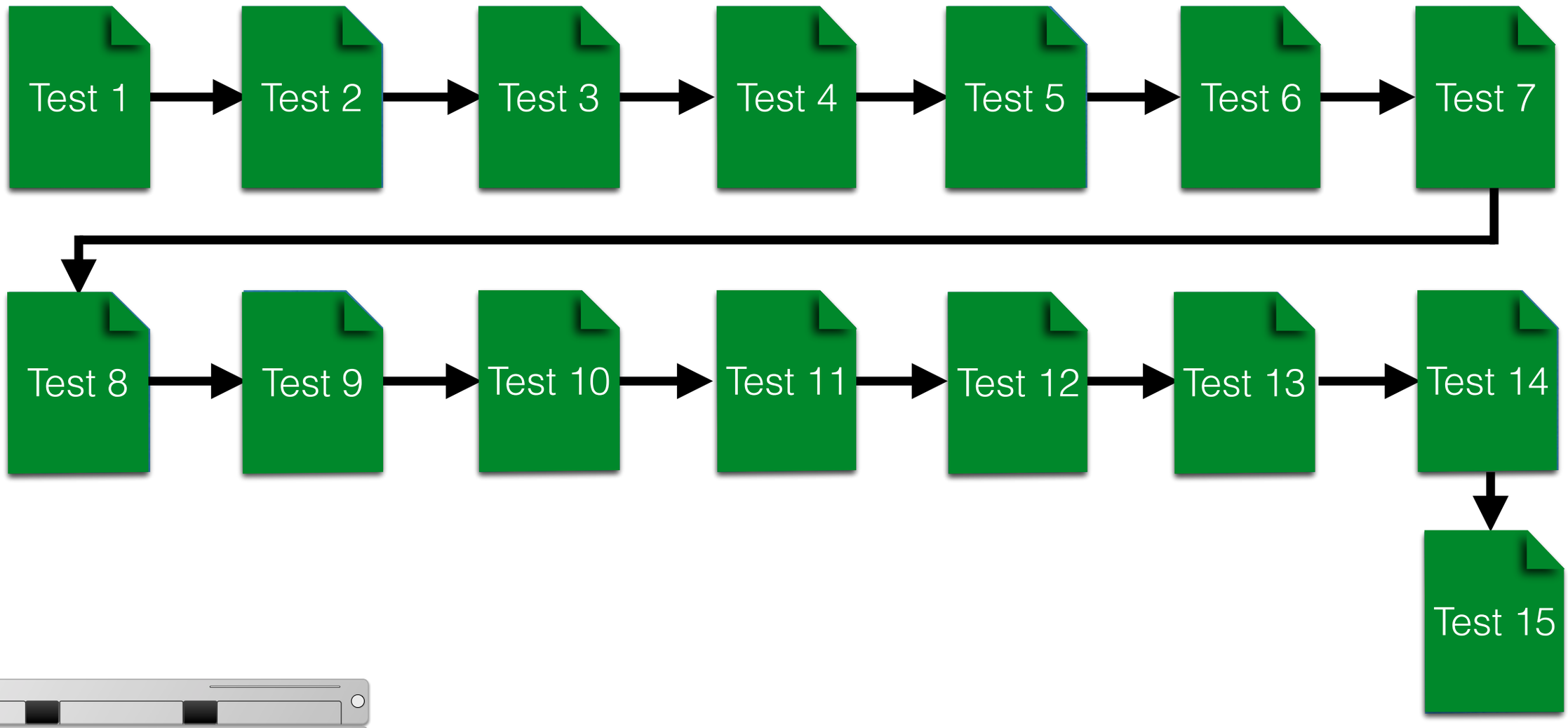
ElectricTest enables sound exploitation of existing test acceleration techniques

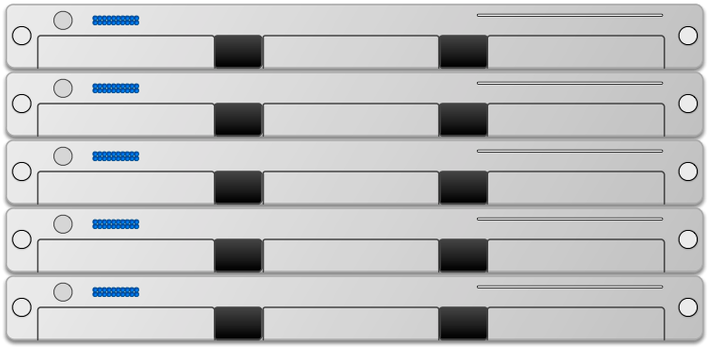
Test Parallelization



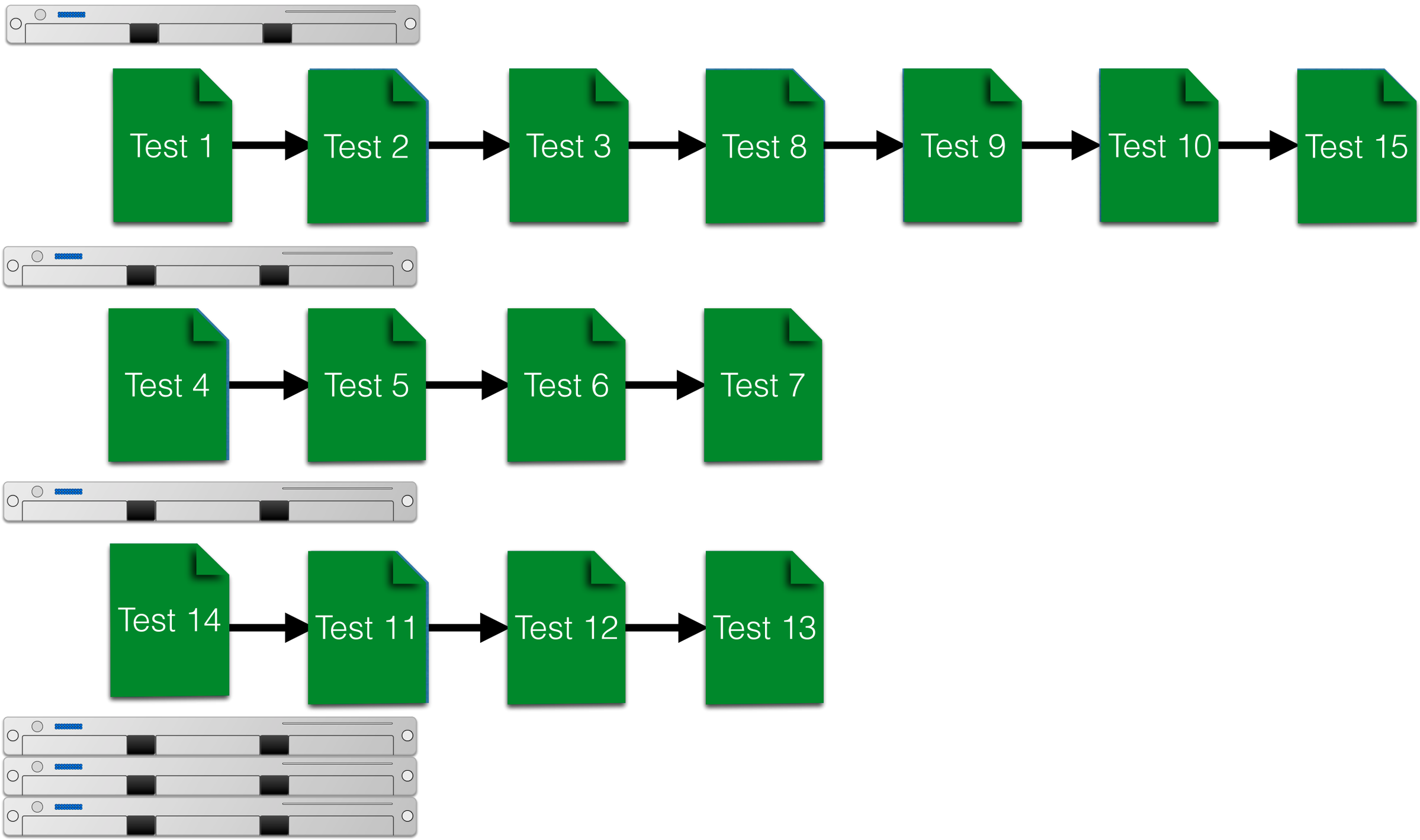
ESEC/FSE @_jon_bell_ September 4, 2015

Safe Test Parallelization

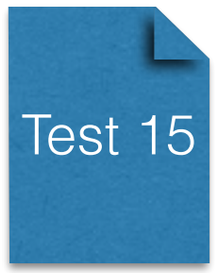




Safe Test Parallelization

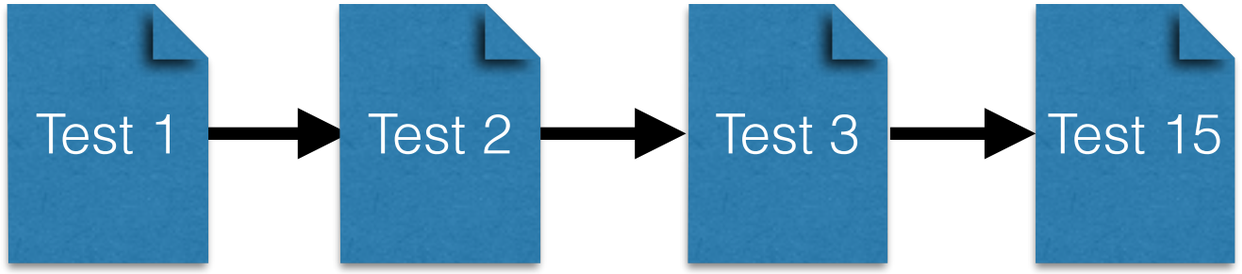


Safe Test Selection



Single test selected to be executed

Safe Test Selection

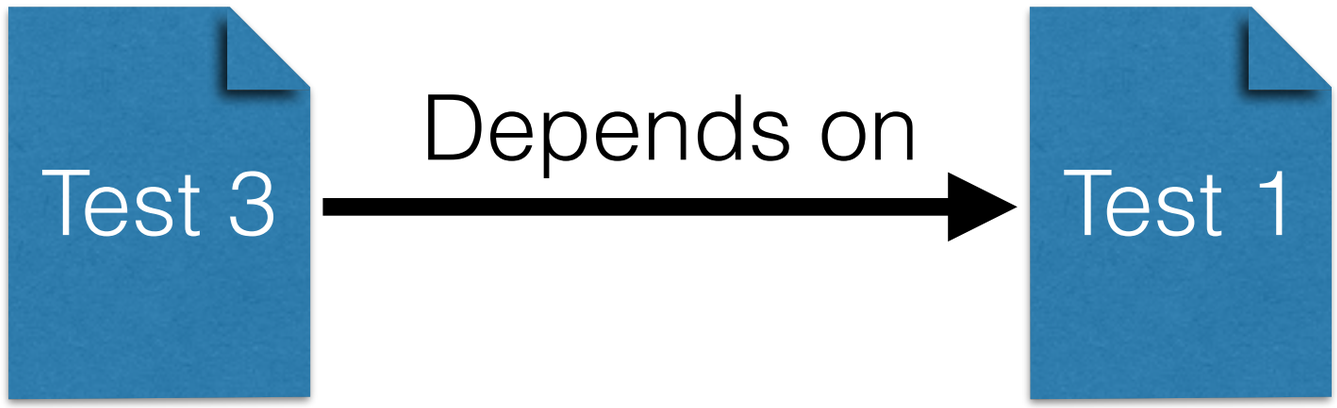


Single test selected to be executed with its dependencies

Understanding Dependencies

- What should a developer do about test dependencies?
  - Might be intentional (e.g. cache shared state)
  - Might be unintentional but OK (e.g. loggers)
  - Might be unintentional and bad (e.g. bug)

Assisting Debugging



Debugging information reported by the previous technique

Assisting Debugging

Value that is read

Test that wrote value

```
Exception in thread "main"
edu.columbia.cs.psl.testdepends.DependencyException: Static Field
ClassA.FieldA member was previously written by Test 1, read here.
    at edu.columbia.cs.psl.testdepends.test.Example$NestedExample.dragons(Example.java:20)
    at edu.columbia.cs.psl.testdepends.test.Example.moreMagic(Example.java:12)
    at edu.columbia.cs.psl.testdepends.test.Example.magic(Example.java:8)
    at edu.columbia.cs.psl.testdepends.test.Example.main(Example.java:15)
```

Stack trace shows use

Evaluation

- RQ1: Recall (accuracy)
- RQ2: Runtime overhead
- RQ3: Impact on acceleration

RQ1: Recall

Project	Dependencies Detected		
	Ground Truth	ElectricTest	
		Writers	Readers
Joda	2	15	121
XMLSecurity	4	3	103
Crystal	18	15	39
Synoptic	1	10	117

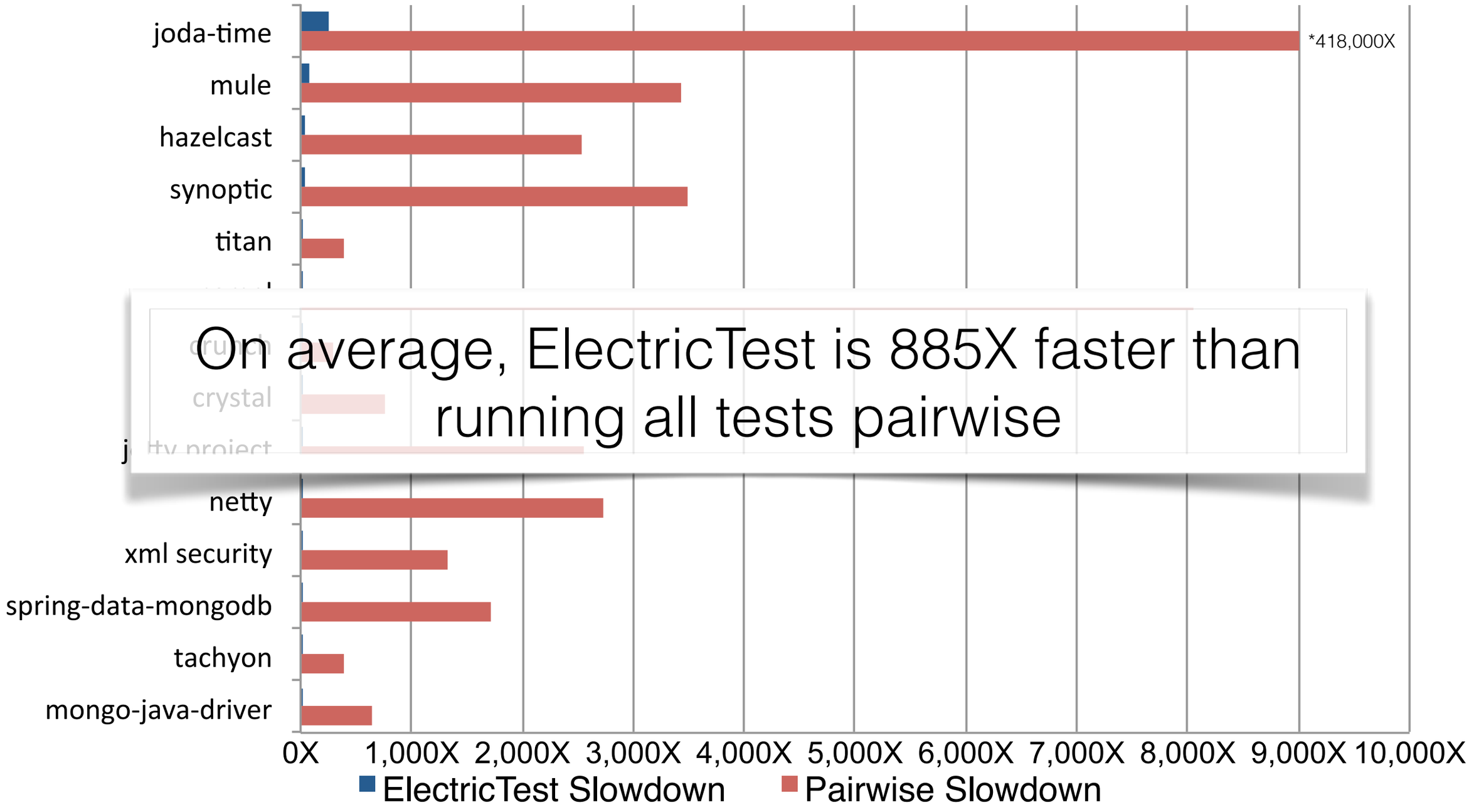
RQ1: Recall

Project	Dependencies Detected			ElectricTest Shared Resource Locations	
	Ground Truth	ElectricTest			
		Writers	Readers	App	Library
Joda	2	15	121	39	12
XMLSecurity	4	3	103	3	15
Crystal	18	15	39	4	19
Synoptic	1	10	117	3	14

RQ2: Overhead

- Selected 10 projects with > 10 minutes of tests
- Also included projects studied by Zhang et al, averaging < 10 seconds of testing
  - Previous exhaustive approach slowdown: $>10^{300}$X
  - Previous heuristic approach slowdown: 31,882X
  - ElectricTest slowdown: 36X (885X faster than previous approach)

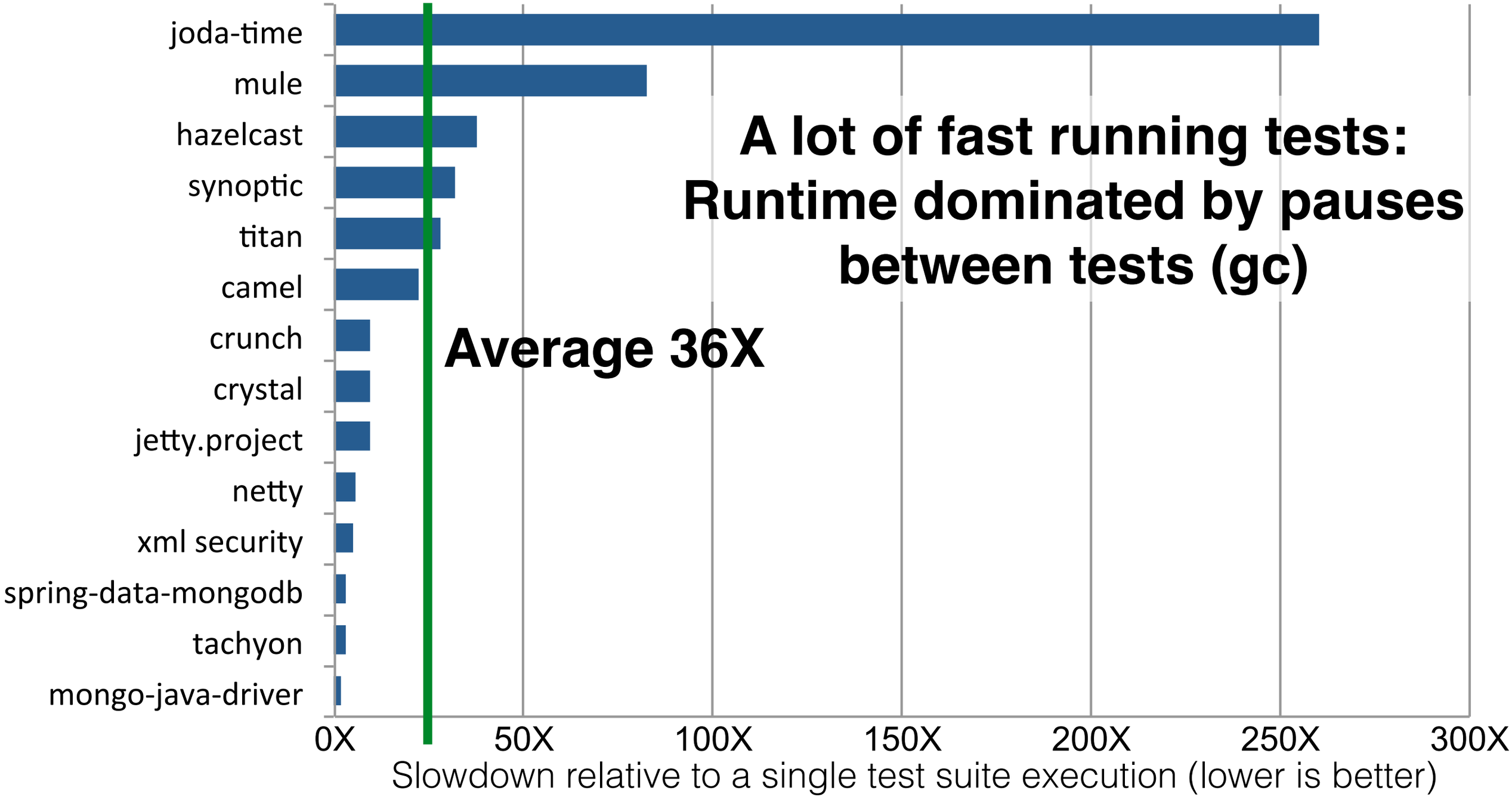
RQ2: Overhead



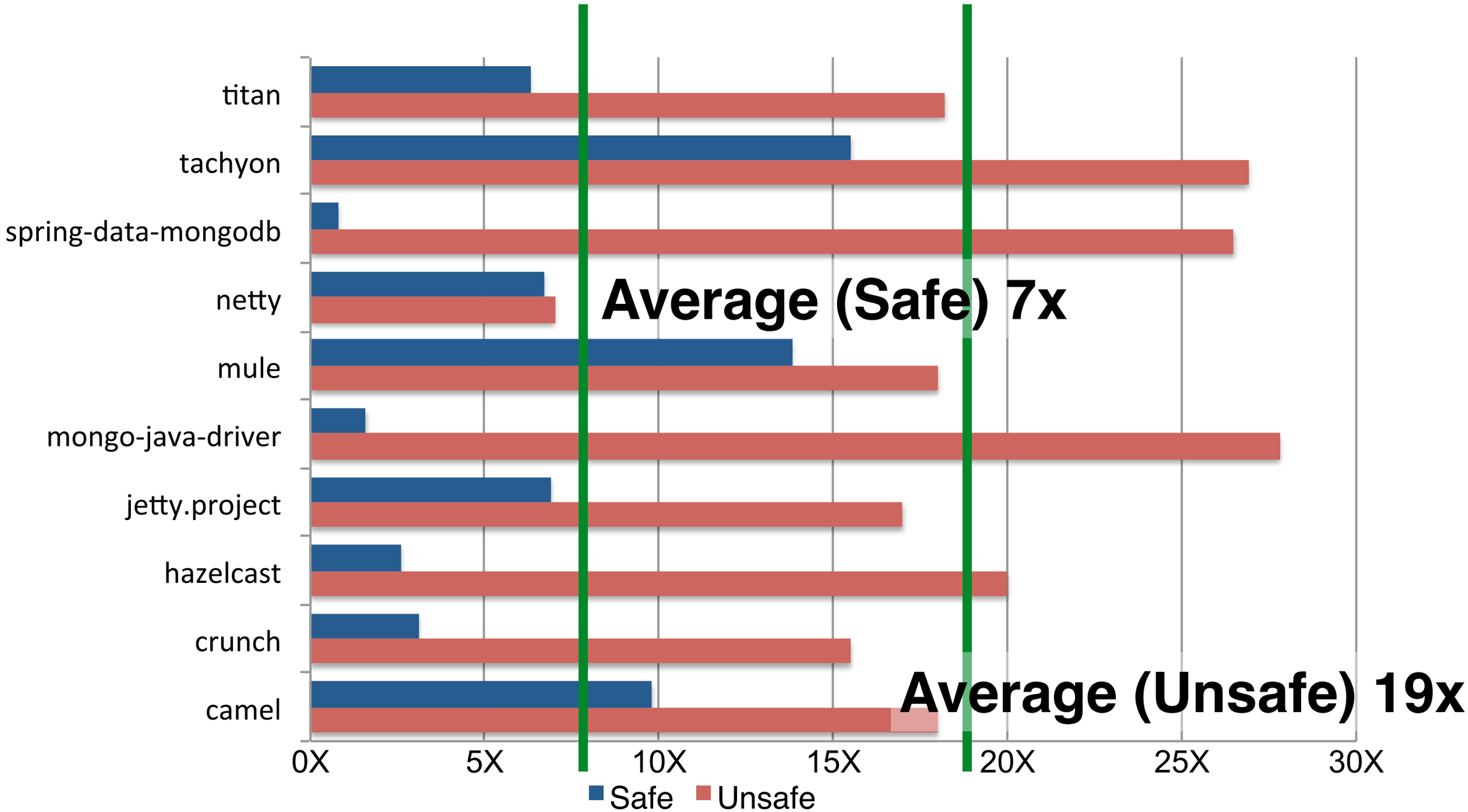
On average, ElectricTest is 885X faster than running all tests pairwise

Slowdown relative to a single test suite execution (lower is better)

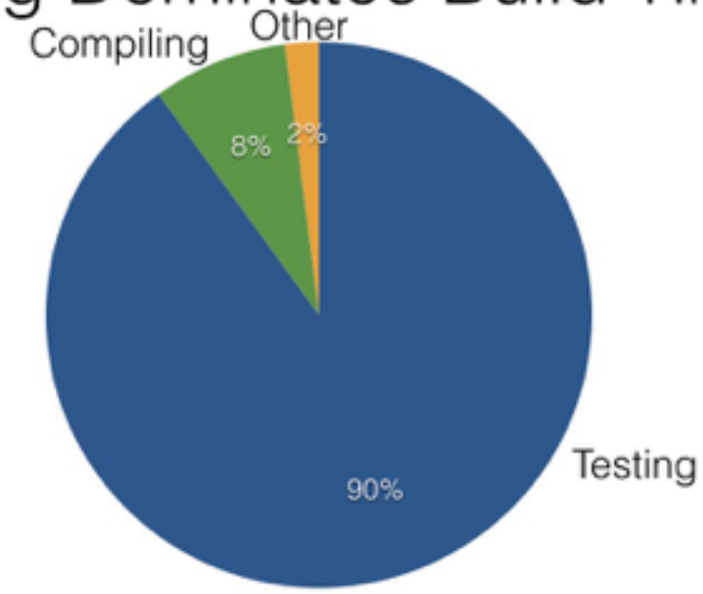
RQ2: Overhead



RQ3: Impact on Acceleration

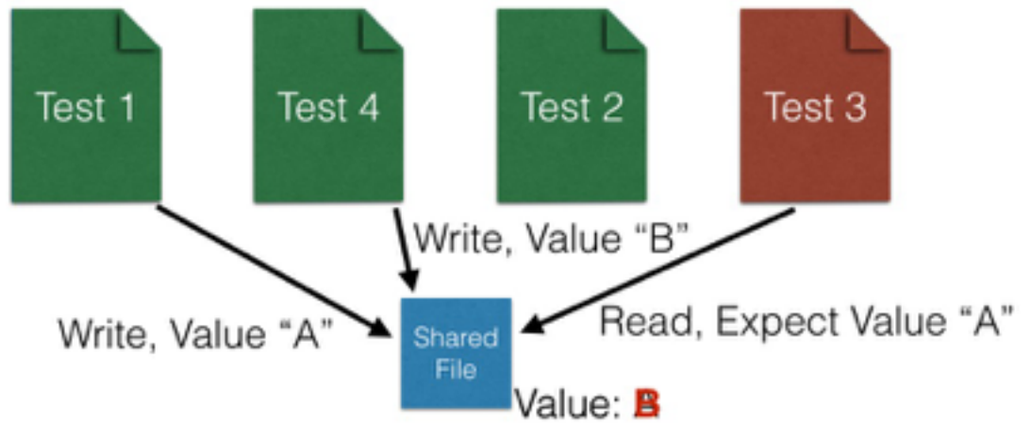


Testing Dominates Build Times

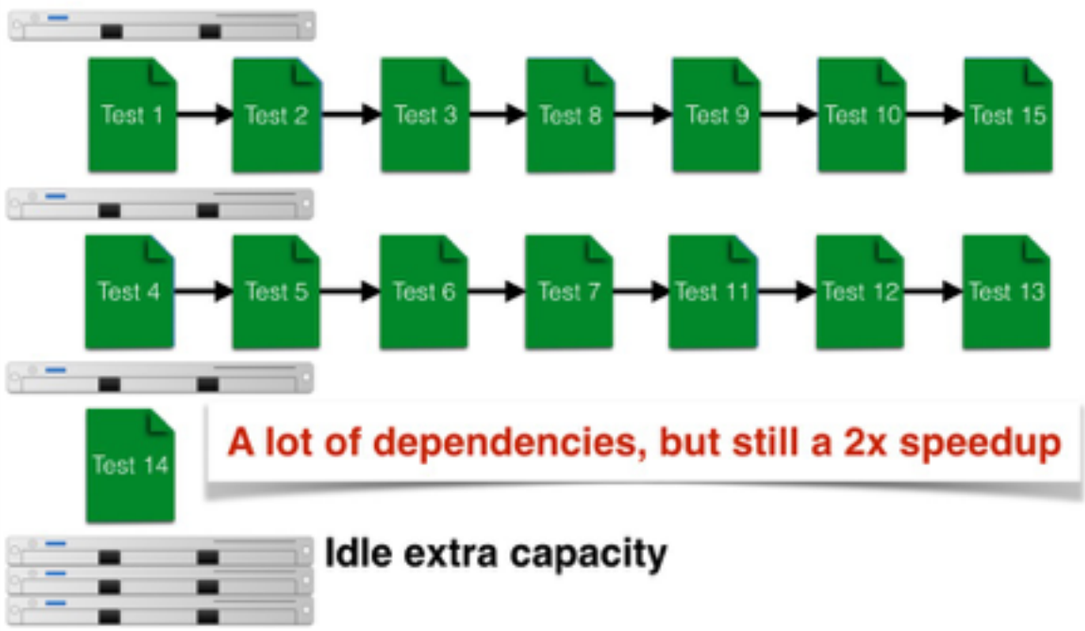


Projects taking > 1 hour to build (8)

Test Dependencies



Intuition

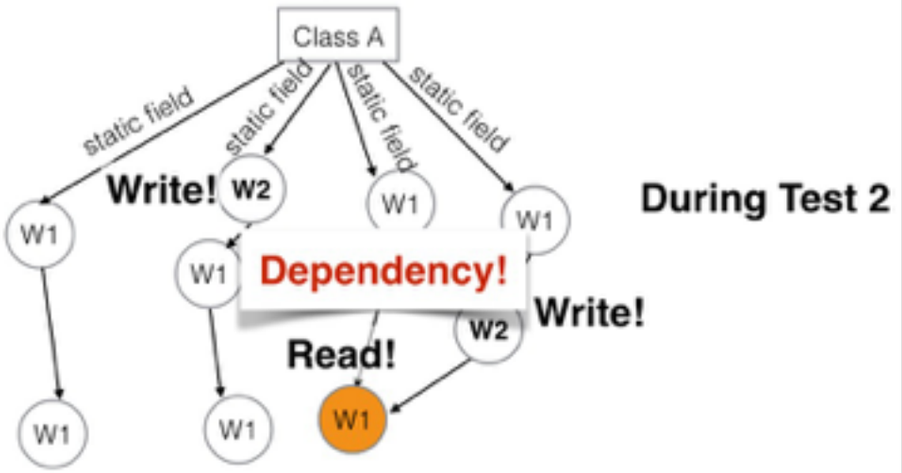


A lot of dependencies, but still a 2x speedup

Idle extra capacity

Efficiently Identifying Dependencies

During test execution, monitor accesses to existing objects



Efficient Dependency Detection for Safe Java Test Acceleration

Jonathan Bell, Gail Kaiser, Eric Melski and Mohan Dattatreya
Columbia University & Electric Cloud, Inc