GRADLE 4.0

ETIENNE STUDER, VP OF PRODUCT TOOLING, GRADLE INC.

Etienne Studer

C: Gradle Inc.
P: Gradle Enterprise

E: etienne@gradle.com
T: @etiennestuder
G: github.com/etiennestuder

4.0M

DOWNLOADS/
MONTH

#17

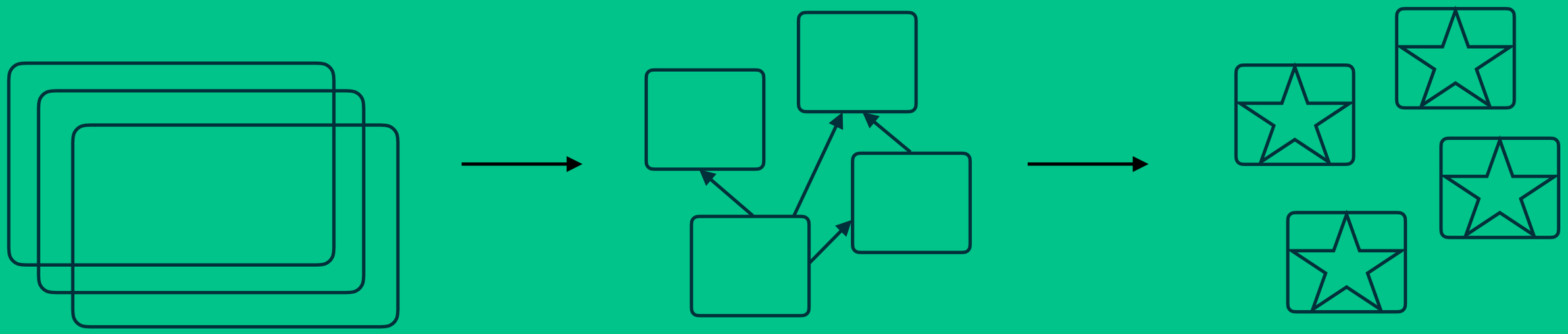
OSS PROJECTS
WORLD-WIDE

30

GRADLE
ENGINEERS

Gradle 1-min intro

Gradle build execution



Gradle build scripts Gradle tasks Gradle task execution

2-phase build:

Configuration phase → build task graph (DAG)

Execution phase → execute task graph

Gradle 1-min intro

Demo

Incremental builds

Observation

Typically, not much changes in the build between consecutive invocations of the build.

When little changes in the build, little work should be done by the build.

Reuse outcomes of the previous run.

Task inputs and outputs

Only run a task if its input or outputs have changed since the previous run.

Inputs —> Task —> Outputs

Example for Compile task:

- Task inputs: source files, compiler flags, etc.
- Task output: class files

Incremental builds

Demo

Build cache

Build cache

- Reuse outcomes of any previous run (rather than just the last)
- Local cache and remote cache
- Task output caching

Build cache

Calculate cache key from inputs, use output as cache value

Inputs —> Task —> Output

Example for Compile task:

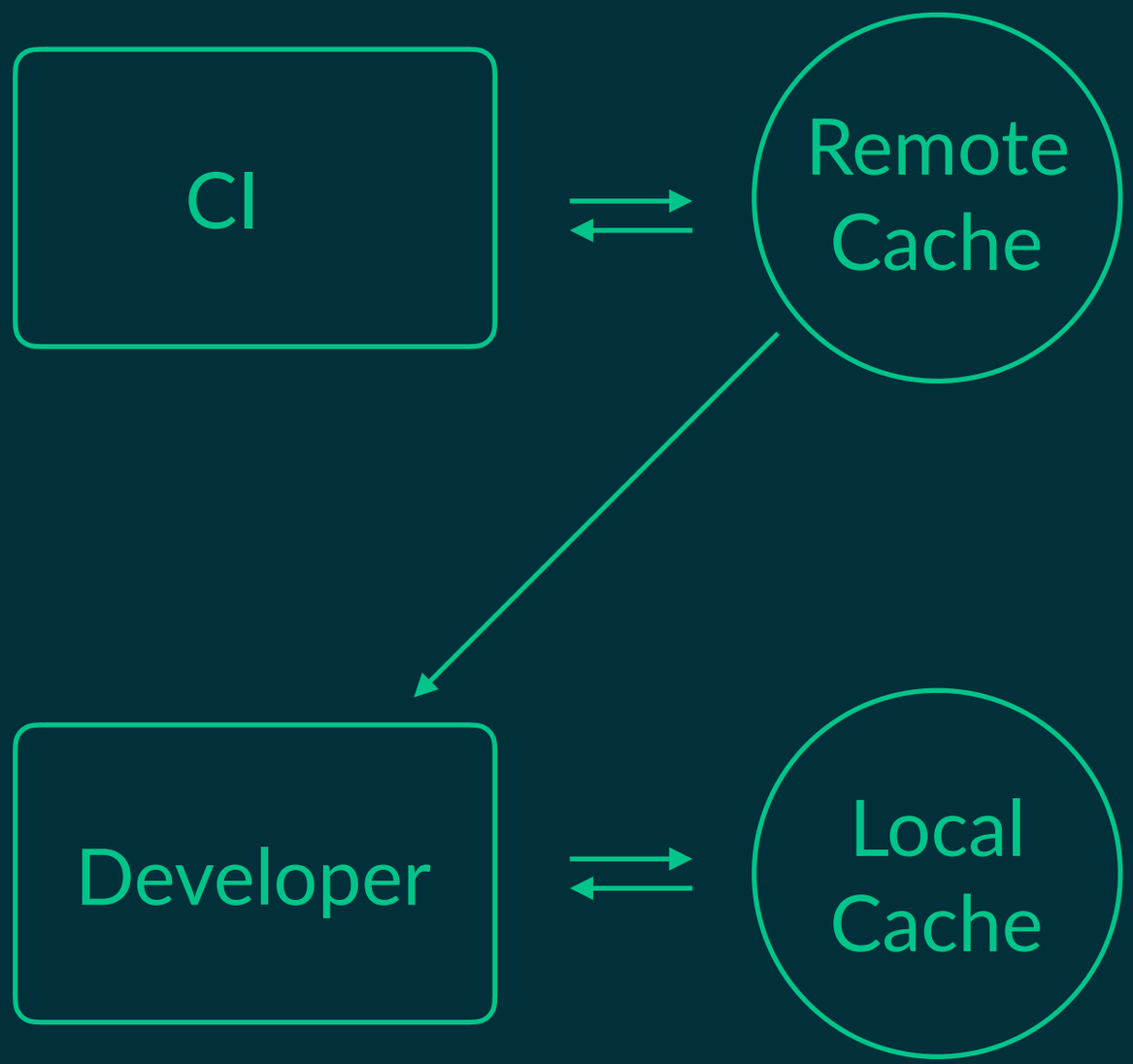
Cache key: hash(source files, compiler flags, etc.)

Cache value: fileTree(class files)

Build cache

Demo

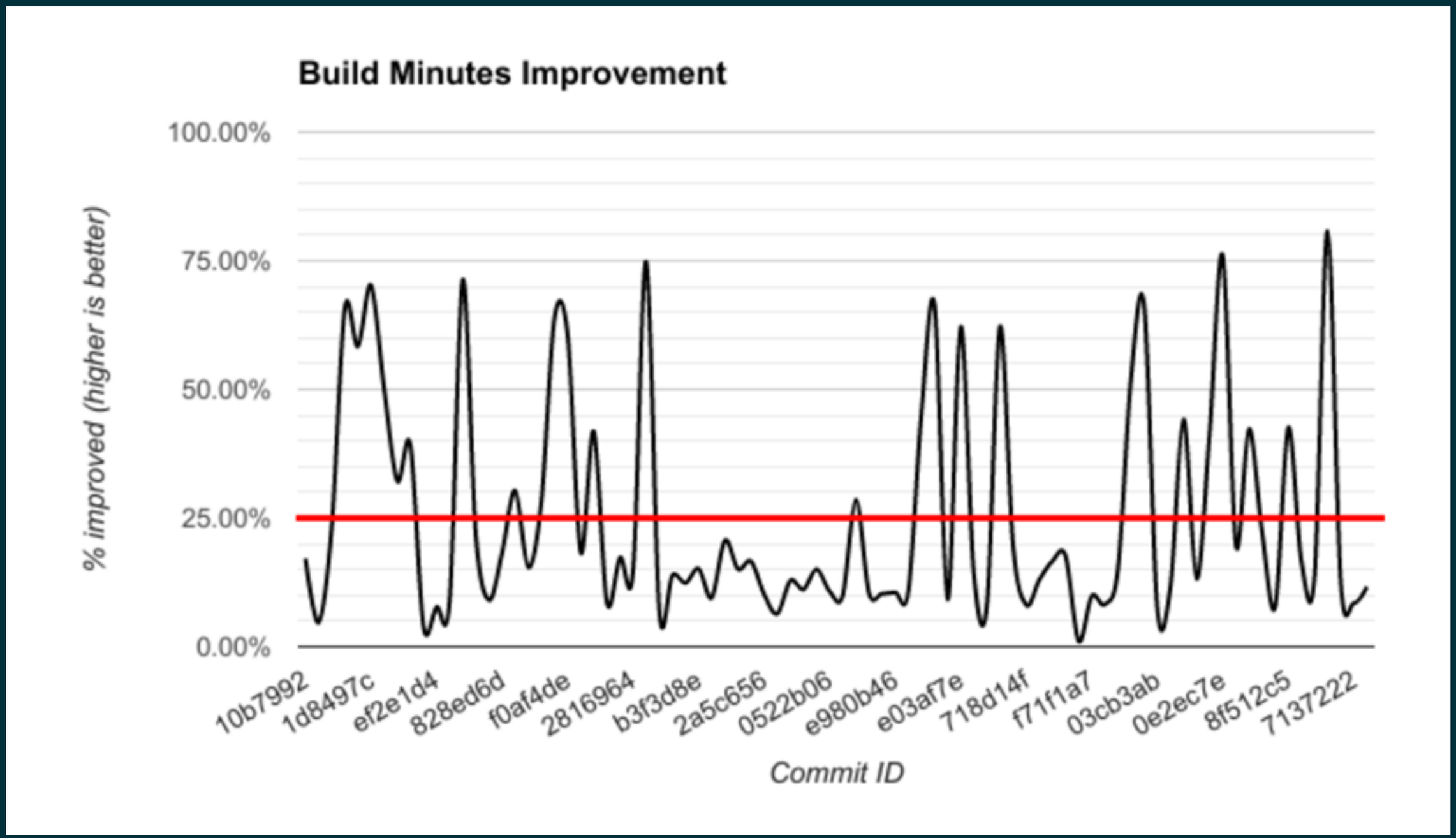
Build cache



Build cache

```
buildCache {
    local {
        enabled = !isCI
    }
    remote(HttpBuildCache) {
        url = ‘https://my.ge.server/cache/'
        push = isCI
    }
}
```

Build cache


Build Minutes Improvement
100.00%
75.00%
50.00%
25.00%
0.00%
% improved (higher is better)
10b7992
1d8497c
ef2e1d4
828ed6d
f0af4de
2816964
b3f3d8e
2a5c656
0522b06
e980b46
e03af7e
718d14f
f71f1a7
03cb3ab
0e2ec7e
8f512c5
7137222
Commit ID

Build cache

Android

- Gradle tasks of Android plugin will become cacheable
- Caching transformed dependencies possibly in the future

Build cache

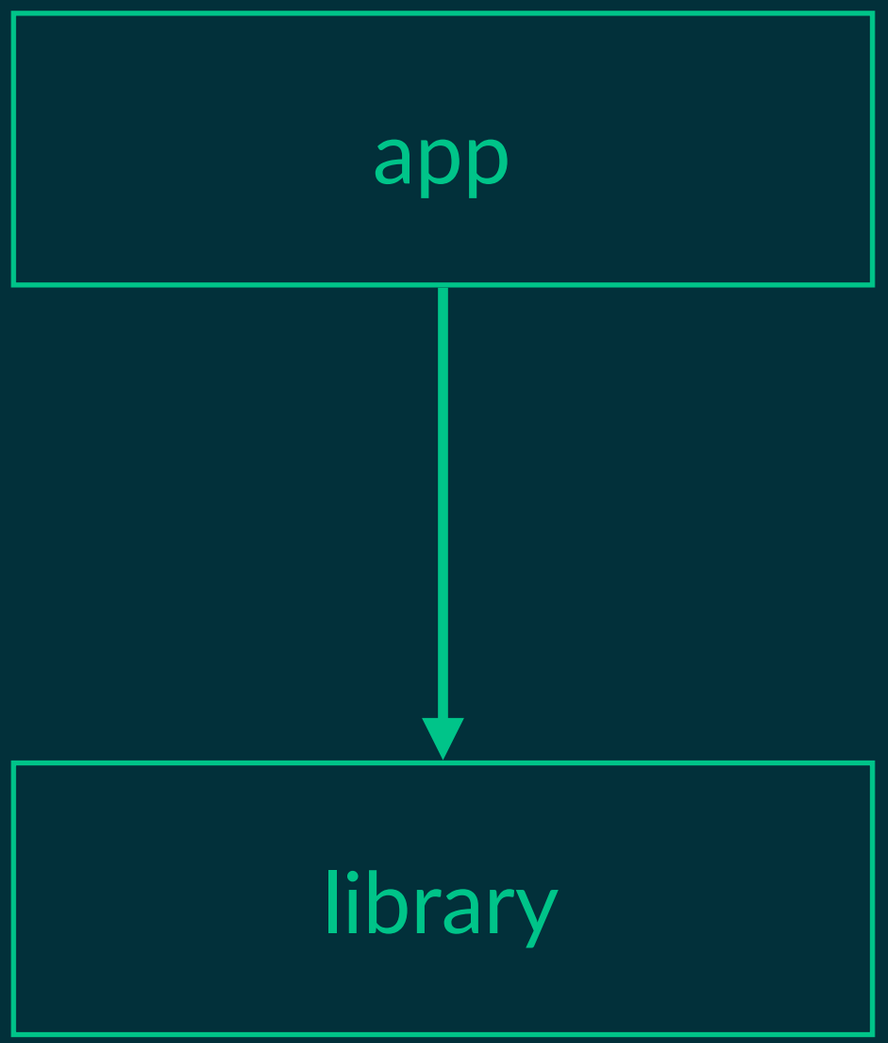
- https://blog.gradle.org/introducing-gradle-build-cache
- https://docs.gradle.org/3.5/userguide/build_cache.html
- High-performant, scalable build cache implementation available in Gradle Enterprise
  - https://gradle.com/build-cache

Compile avoidance & incremental compiler

Compile avoidance & incremental compiler

Avoid wasting time compiling source classes that do not have to be compiled.

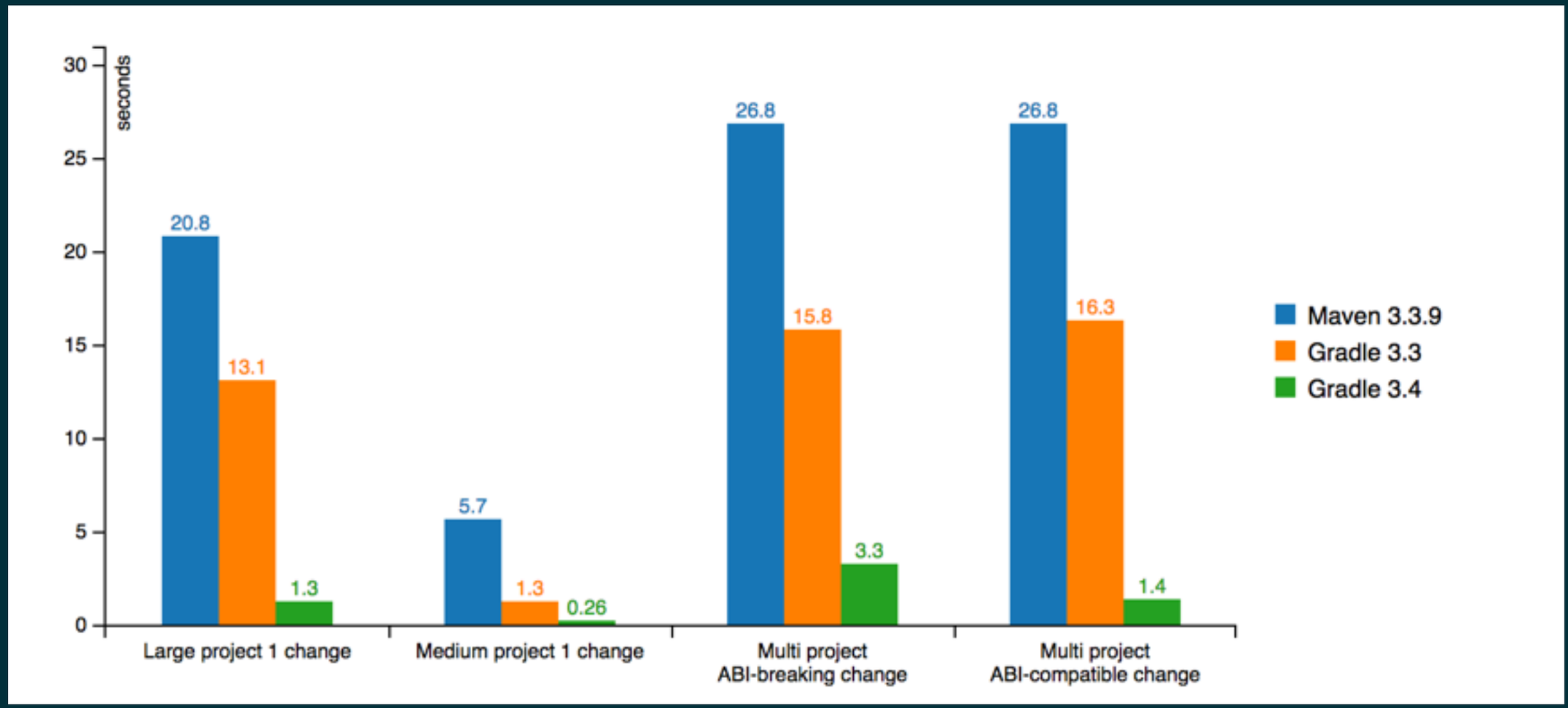
Compile avoidance & incremental compiler



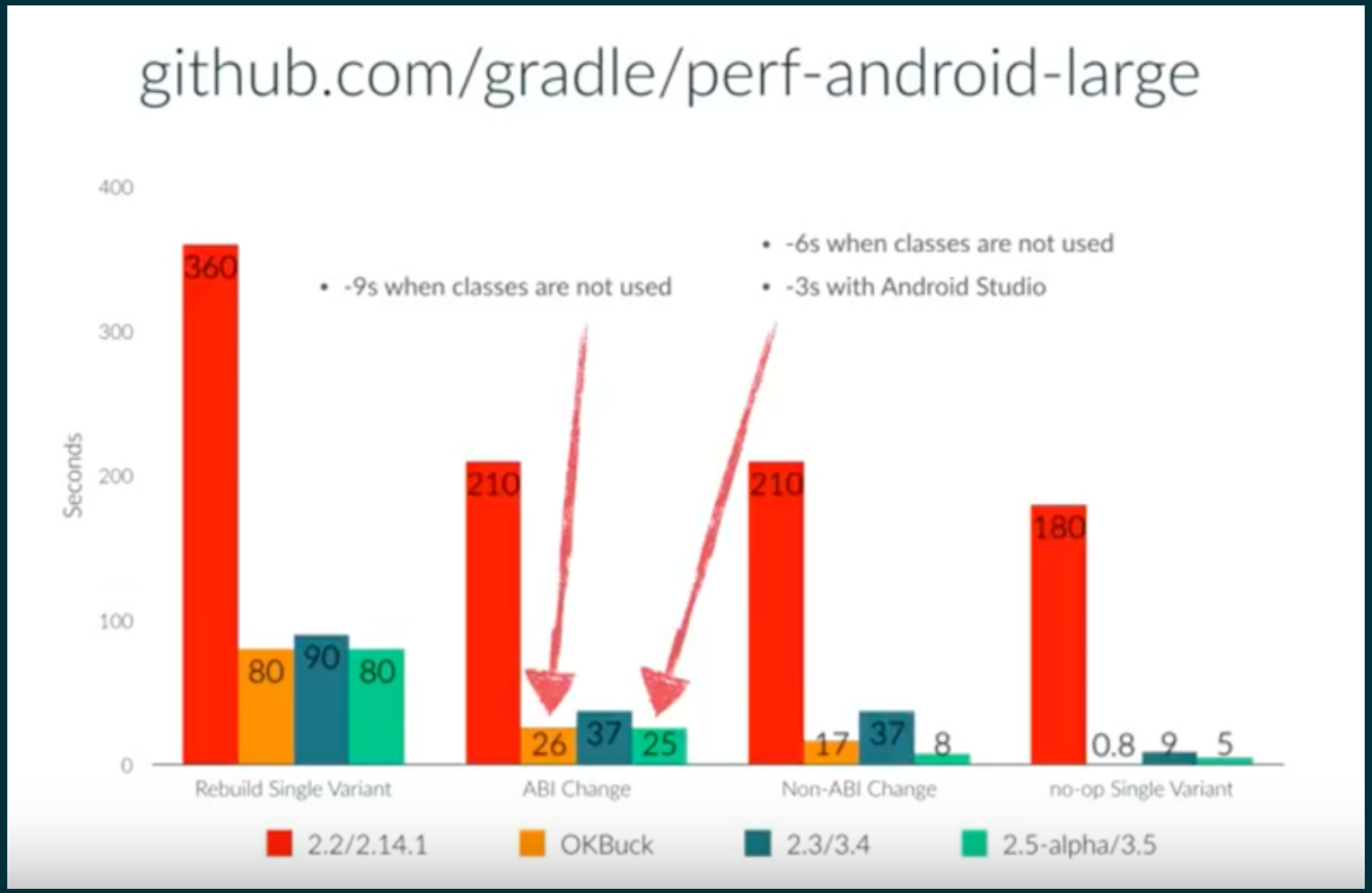
Compile avoidance & incremental compiler

Demo

Compile avoidance & incremental compiler



Compile avoidance & incremental compiler



Compile avoidance & incremental compiler

- https://blog.gradle.org/incremental-compiler-avoidance
- https://docs.gradle.org/current/userguide/java_plugin.html

General performance improvements

General performance improvements

- Faster configuration time
- Parallel dependency resolution
- Parallel task / action execution by default

Gradle daemon

Gradle daemon

Gradle builds executed much more quickly by a long-lived background process that avoids expensive bootstrapping and leverages caching.

Gradle daemon

Demo

Gradle daemon

- https://docs.gradle.org/current/userguide/gradle_daemon.html

Worker API

Worker API

- API to run task actions in parallel safely
- Parallel actions cannot mutate shared state
- Supports out-of-process and in-process actions

Continuous build

Continuous build

Demo

Continuous build

- https://blog.gradle.org/introducing-continuous-build
- https://docs.gradle.org/current/userguide/continuous_build.html

Composite builds

Composite builds

- Fix a bug in a library through app using project
- Break down a monolith into multiple repos
- Consume latests state of libraries in integrations builds

Composite builds

Demo

Composite builds

- https://blog.gradle.org/introducing-composite-builds
- https://blog.jetbrains.com/idea/2017/03/webinar-recording-composite-builds-with-gradle
- https://docs.gradle.org/current/userguide/composite_builds.html

New Gradle console

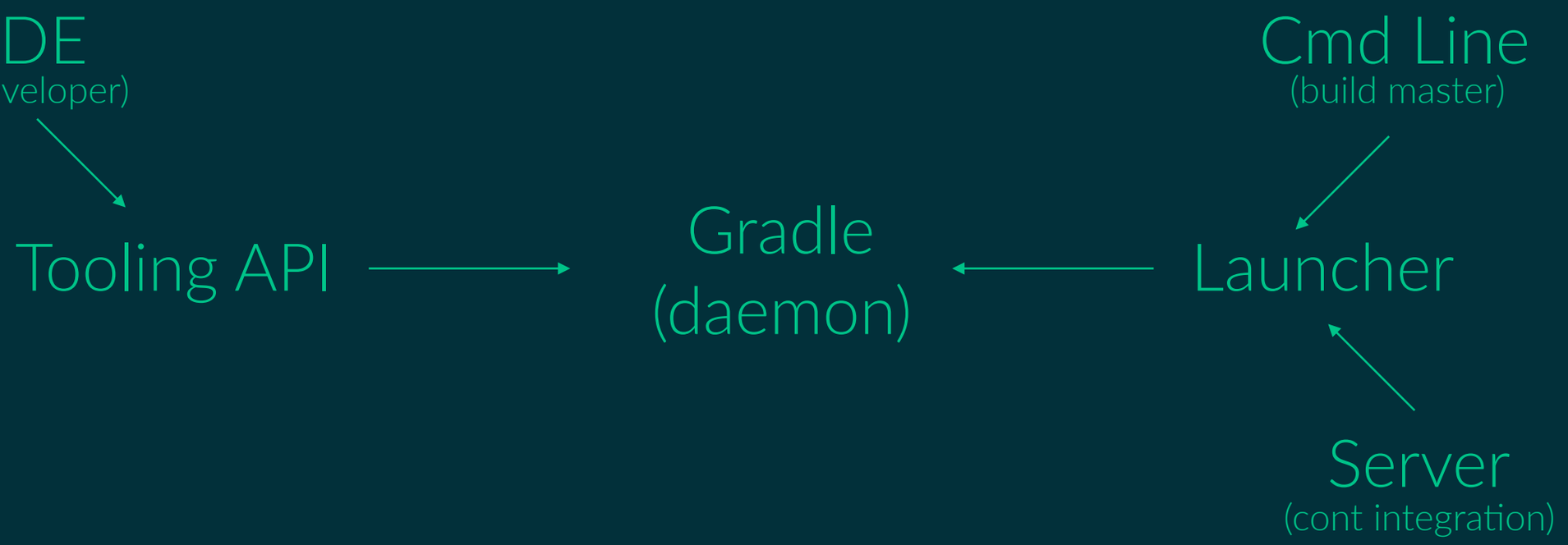
IDE integration

IDE integration

- Project import & synchronization
- Task execution
- Test execution
- Build execution insights

Build is the single source of truth!

Tooling API


IDE
(developer)
Cmd Line
(build master)
Tooling API
Gradle
(daemon)
Launcher
Server
(cont integration)

IDE integration

Demo

IDE integration

- https://blog.gradle.org/announcing-buildship-2.0
- https://marketplace.eclipse.org/content/buildship-gradle-integration

Gradle Script Kotlin

Gradle Script Kotlin

- Syntax highlighting
- Quick documentation
- Navigation to source
- Auto-completion / content assist
- Refactoring

- High execution time performance

Build code is no different to application code!

Gradle Script Kotlin

- https://blog.gradle.org/kotlin-meets-gradle
- https://github.com/gradle/gradle-script-kotlin

Gradle Profiler

Gradle Profiler

- Profiling and benchmarking
- DSL to describe scenarios (tasks, Gradle version, etc.)
- Profiles for Build scans, YourKit, Chrome Trace, etc.
- Build life-cycle insights derived from build operations

Gradle Profiler

- github.com/gradle/gradle-profiler

Build scans

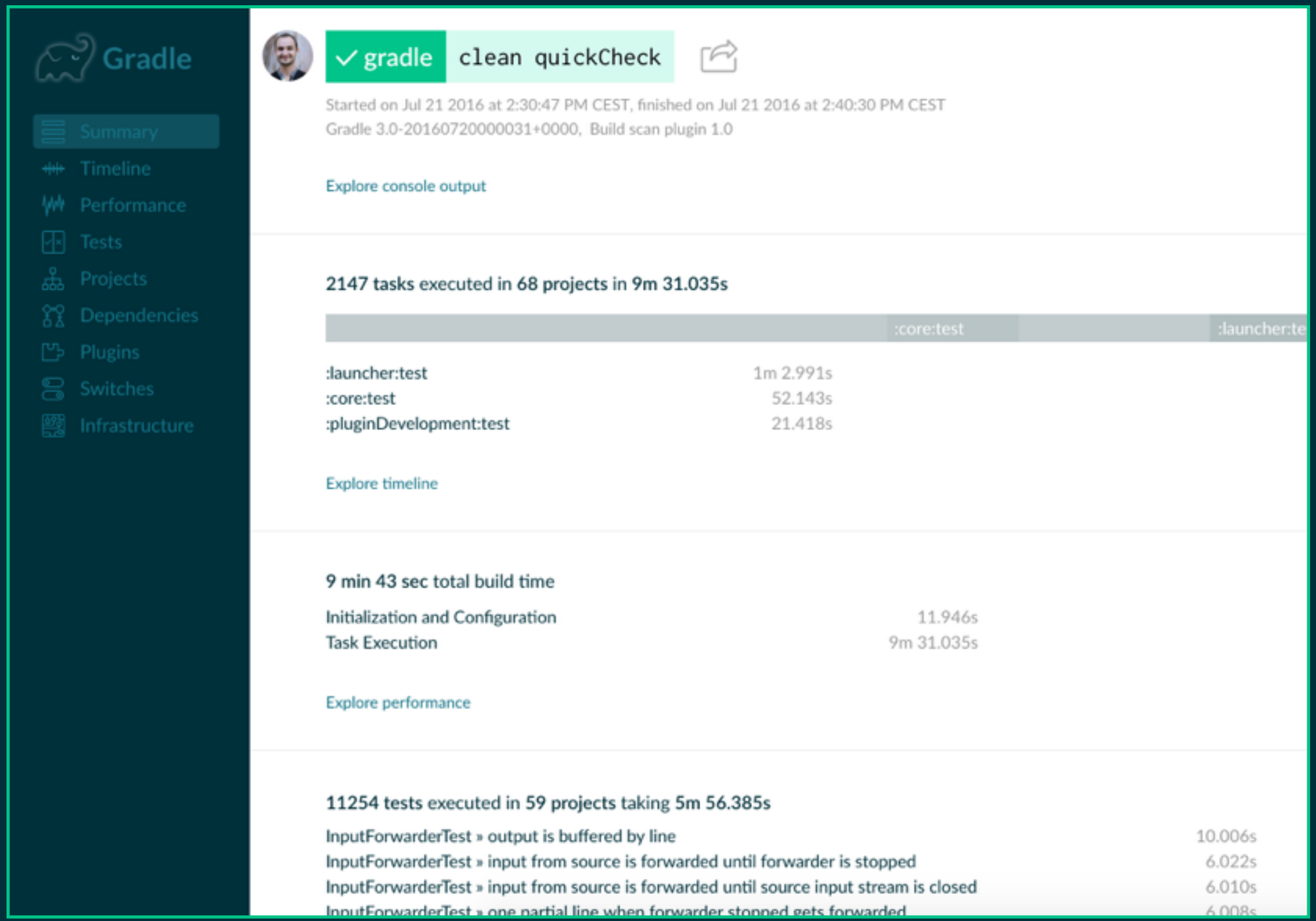
Build scans

- Gain build insights
- Improve build performance
- Collaborate with colleagues and the community

Build scans

Demo

Build scans


Gradle
Summary
Timeline
Performance
Tests
Projects
Dependencies
Plugins
Switches
Infrastructure
gradle clean quickCheck
Started on Jul 21 2016 at 2:30:47 PM CEST, finished on Jul 21 2016 at 2:40:30 PM CEST
Gradle 3.0-20160720000031+0000, Build scan plugin 1.0
Explore console output
2147 tasks executed in 68 projects in 9m 31.035s
:core:test
:launcher:test 1m 2.991s
:core:test 52.143s
:pluginDevelopment:test 21.418s
Explore timeline
9 min 43 sec total build time
Initialization and Configuration 11.946s
Task Execution 9m 31.035s
Explore performance
11254 tests executed in 59 projects taking 5m 56.385s
InputForwarderTest » output is buffered by line 10.006s
InputForwarderTest » input from source is forwarded until forwarder is stopped 6.022s
InputForwarderTest » input from source is forwarded until source input stream is closed 6.010s

Build scans

- https://gradle.com
- https://gradle.com/scans/get-started

Build scans are a free service for everyone!

Gradle Enterprise

Gradle Enterprise

- Query scans
- Compare scans
- Use a high-performance, scalable build cache
- Host within your firewall

Gain deep build insights within your company and teams!

Gradle Enterprise

Demo

Gradle Enterprise

- https://gradle.com/enterprise

Gradle Enterprise is a commercial offering!

Android

Android plugin 2.5

- Close collaboration between Gradle and Google Android Team

Android plugin 2.5

- No more dependency resolution at configuration time
- Parallel dependency resolution (artifacts / metadata download)
- Dependencies always only downloaded once per build
- Variant-aware dependency management
- More fine-grained parallelism
- Compile avoidance
- Explicit annotation-processor declaration
- Linear growth when new variants are added
- Android tasks will be cacheable

Gradle

- Incremental builds
- Build cache
- Compile avoidance & incremental compiler
- Worker API
- Daemon
- Continuous builds
- Composite builds
- Tooling API
- Statically-typed DSL
- Build scans
- Gradle Enterprise

Used by LinkedIn NETFLIX unity

Thank you

Etienne Studer