Joker<?> 16

Gradle 3.0 and beyond

René Gröschke

WHO AM I

```
speaker {
    name 'René Gröschke'
    homebase 'Berlin, Germany'
    work 'Principal Engineer @ Gradle Inc.'
    twitter '@breskeby'
    github 'breskeby'
    email 'rene@gradle.com'
    hobbies '...'
}
```

WHAT IS GRADLE?

EMERGE FROM BUILD HELL

- Unified, fast, reproducible builds
- Cross-platform
- Language agnostic
- A Build Tool + Cloud Services

IN A NUTSHELL

A simple java project

```
apply plugin:"java"

version = file("version.txt").text

repositories {
    jcenter()
}

dependencies {
    testCompile "junit:junit:4.+"
}

task printVersion << { println "We're using - version '$version'!" }
```

NO BIG BANG RELEASE

- 3.0 released on August 15th 2016
- 3.1 released on September 19th 2016
- 3.2 RC-1 to be expected next week
- Releasing every 4 - 6 weeks.

A TYPICAL DEVELOPER WORKFLOW

1. Edit codebase
2. Run a build
3. Examine outputs
4. Edit codebase
5. Run a build
6. Examine outputs
7. Edit…
8. Ru…
9. …

LET'S HAVE SOME AUTOMATION FOR THIS

CONTINUOUS BUILDS

- Monitors task inputs / outputs
- Triggers rebuild if any of both has changed
- Provides fast feedback loop

```
gradle build -t
```

TDD ANYBODY?

AMBITIOUS AUTOMATION CAN GET COMPLEX

GRADLE TESTKIT

Functional testing of your build logic

```
apply plugin: 'groovy'

repositories {
    mavenCentral()
}

dependencies {
    testCompile localGroovy()
    testCompile gradleTestKit()

    testCompile('org.spockframework:spock-core:1.0-groovy-2.4') {
        exclude module: 'groovy-all'
    }
}
```

GRADLE TESTKIT

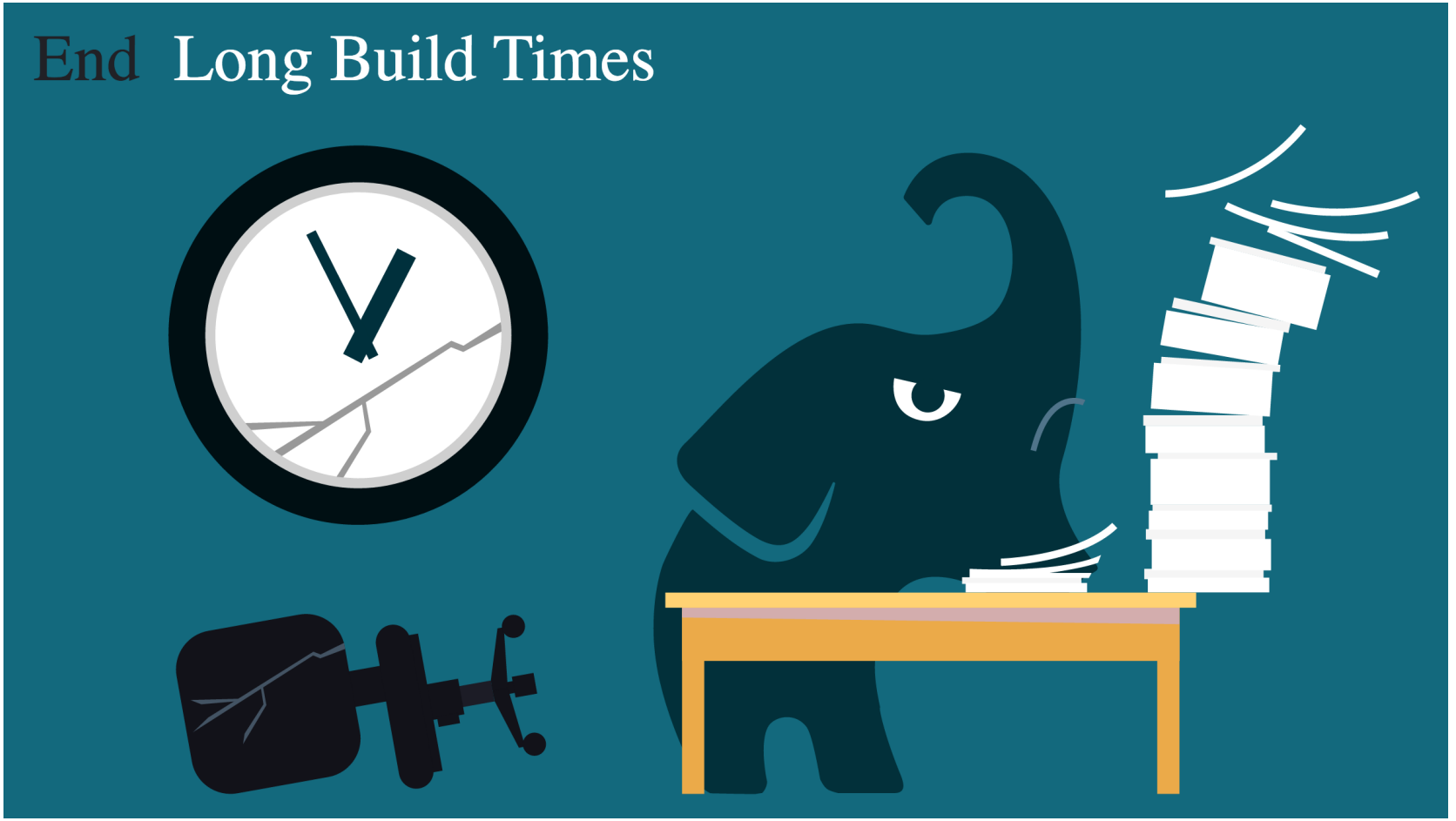
```
def "helloWorld task prints hello world"() {
    given:
    buildFile << """
        task helloWorld << {
            println 'Hello world!'
        }"""

    when:
    def result = GradleRunner.create()
        .withProjectDir(testProjectDir)
        .withArguments('helloWorld')
        .build()

    then:
    result.output.contains('Hello world!')
    result.task(":helloWorld").outcome == SUCCESS
}
```

GRADLE TESTKIT

Demo

WHOSE BUILD IS TOO FAST?

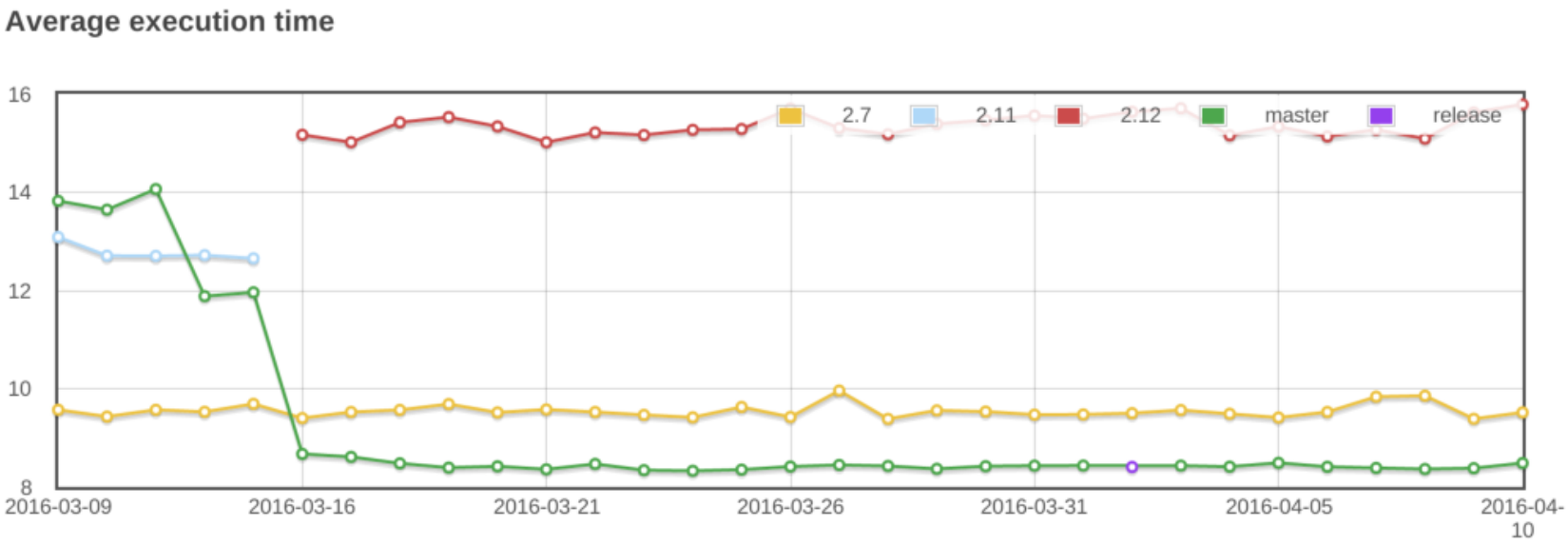
PERFORMANCE IS A FEATURE



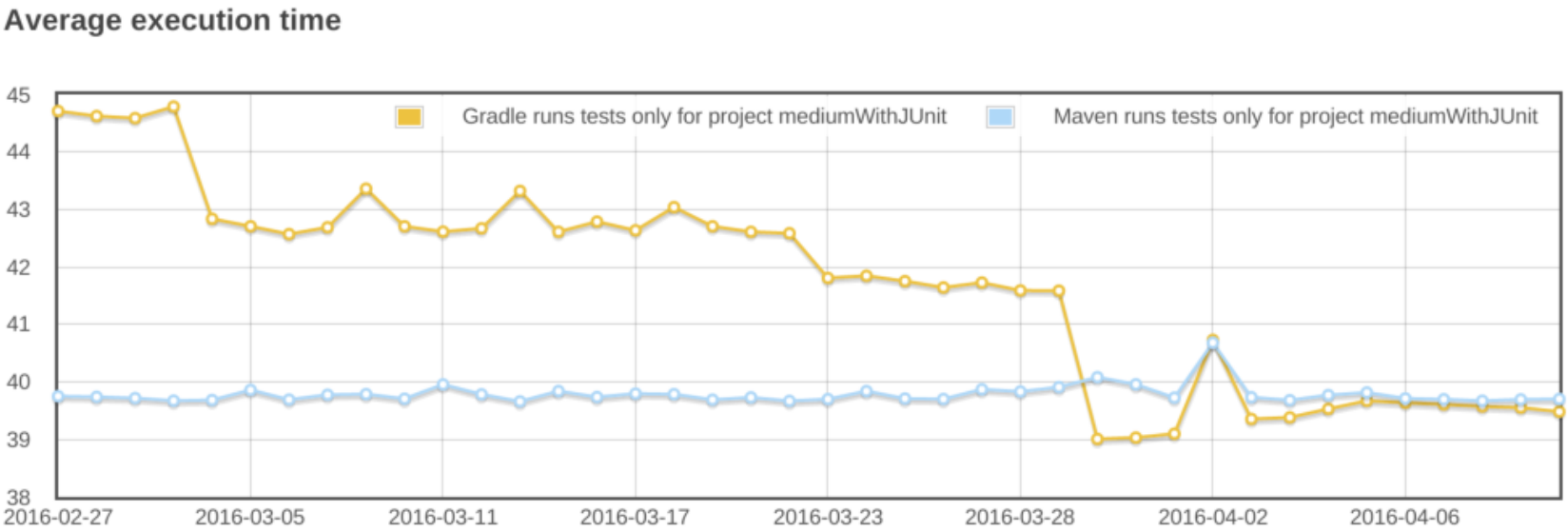
DEDICATED PERFORMANCE BURST

- Reduced configuration time
- Faster script compilation
- Faster up-to-date checks
- Faster test execution
- Faster IDE integration
- Faster dependency resolution
- Ongoing effort

CONFIGURATION TIME SPEEDUP



TEST EXECUTION SPEEDUP



GRADLE DAEMON

- A long-lived background process
- Avoids costly jvm bootstrapping
- Benefits from warmed up hotspot compilation

MORE DAEMON GOODNESS

- On by default since 3.0
- More robust by
  - Dealing better with memory leaks
  - Having clever expiration strategies
- More communicative
- Leveraging daemon more in the future

GRADLE DAEMON

```
# for gradle < 3.0
# ~/.gradle/gradle.properties
org.gradle.daemon=true
```

FOUR FACTS ABOUT KOTLIN

- Statically typed JVM language
- Driven by pragmatism
- Deep support in Idea and Eclipse IDEs
- Allows declarative syntax for crafting DSL with ease.

KOTLIN DSL IN GRADLE

- Working closely with Jetbrains
- M2 support in Gradle 3.0
- v0.3.3 in Gradle 3.2

KOTLIN IN GRADLE

```
apply<ApplicationPlugin>()

configure<ApplicationPluginConvention> {
    mainClassName = "samples.HelloWorld"
}

configure<JavaPluginConvention> {
    setSourceCompatibility(1.7)
}

repositories {
    jcenter()
}

dependencies {
    testCompile("junit:junit:4.12")
}
```

KOTLIN IN GRADLE II

```
val myTask = task("myTask") {
    extra["foo"] = 42
    doLast {
        println("Extra property value: ${extra["foo"]}")
    }
}

afterEvaluate {
    println("myTask.foo = ${myTask.extra["foo"]}")
}

defaultTasks(myTask.name)
```

EVER WISHED TO COMBINE MULTIPLE BUILDS?

COMPOSITE BUILDS

Szczepan Faber
@mockitoguy

New @Gradle is truly disruptive - “composite builds” really means “cross-repo development & refactoring in IDE"!!!
docs.gradle.org/current/releas…

COMPOSITE BUILDS

Defined in a `settings.gradle` file:

```
// settings.gradle
rootProject.name='adhoc'

includeBuild '../my-app'
includeBuild '../my-utils'
```

Or passed via command line argument:

```
> gradle --include-build ../my-utils run
```

COMPOSITE BUILDS

Demo

INCREMENTAL GRADLE BUILDS

- Task up-to-date check been there forever
- Relies on tasks inputs/outputs model

WE ARE REUSING RESULTS…

from **last time**
when we ran **this build**
on **this machine**.

WE CAN DO BETTER

WHY NOT…

from **anytime before**
when we ran **any build**
anywhere.

TASK OUTPUT CACHE

```
> gradle clean logging:assemble
...
:native:classpathManifest
:native:compileJava CACHED
:native:compileGroovy UP-TO-DATE
:native:processResources UP-TO-DATE
:native:classes
:native:jar CACHED
:logging:compileJava CACHED
:logging:compileGroovy UP-TO-DATE
:logging:processResources UP-TO-DATE
:logging:classes
:logging:jar CACHED
:logging:assemble UP-TO-DATE

BUILD SUCCESSFUL
```

TASK OUTPUT CACHE IN ACTION

Demo

TALKING ABOUT BUILDS

WOULDN'T IT BE NICE IF WE COULD…

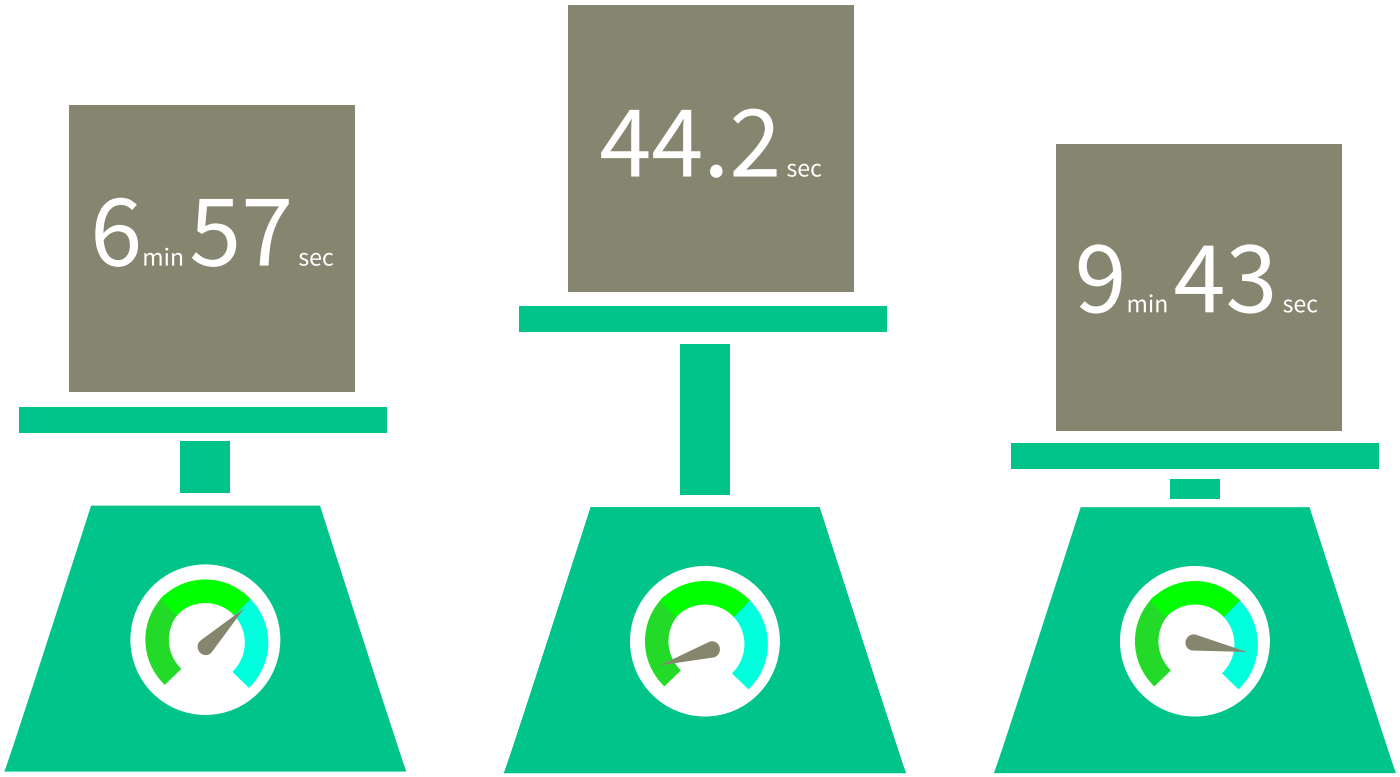
COLLABORATE

…easily share builds to debug issues together?

OPTIMIZE BUILD PERFORMANCE

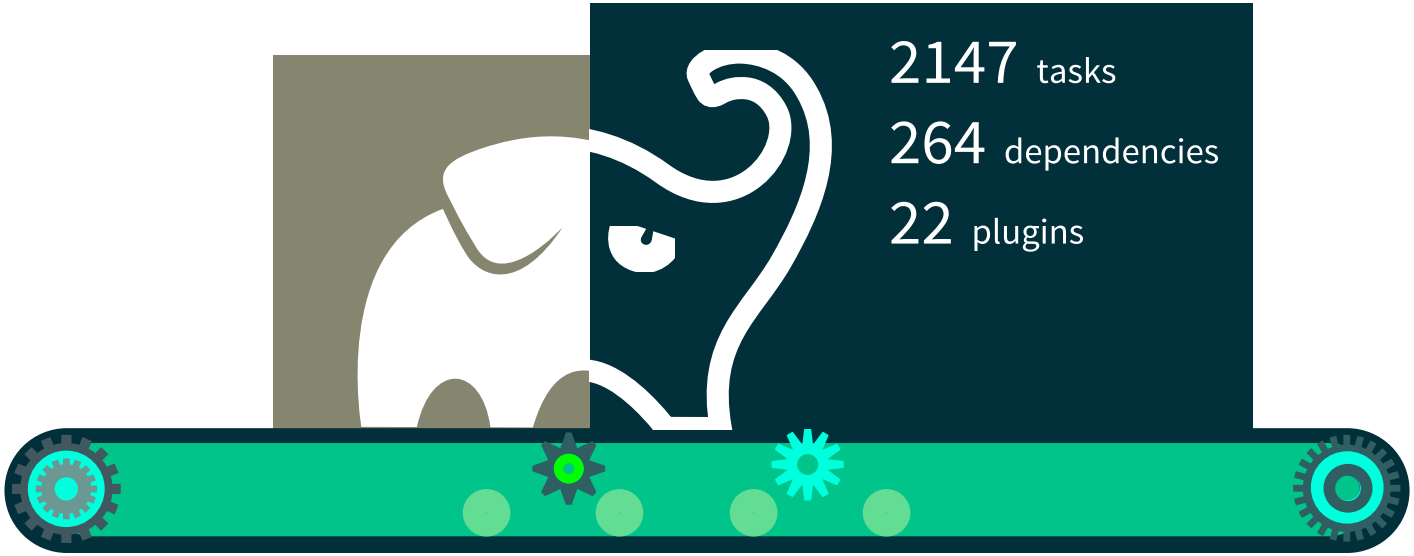
…easily understand where our build time is going and make our builds faster?

COMPARE



Compare builds within our entire organization?

DISCOVER

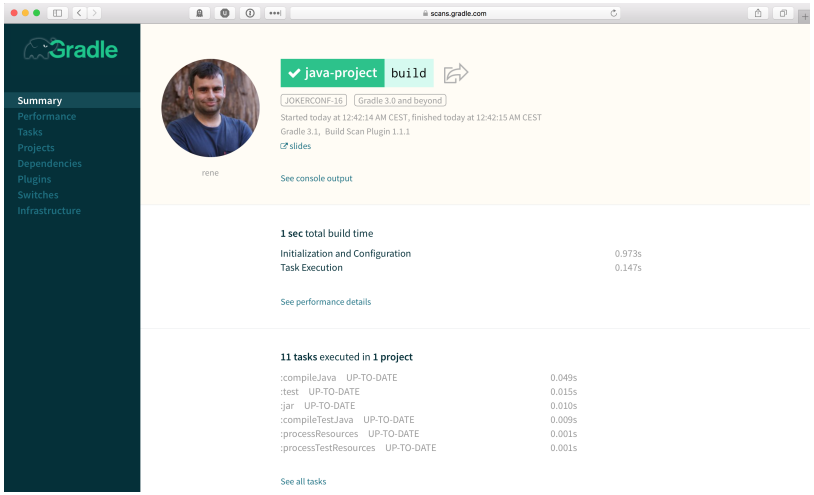


…discover

- how our software is actually being built within our entire organization?
- where our build time is going and make our builds faster?

INTRODUCING GRADLE BUILD SCANS

- Insights into your build
- View and share via URL
- Debug, optimize and refine
- Communicate via builds
- Analyze all of your builds



rnate

5 min 44 sec

asks

dependencies

ugins

JUnit 5

2 min 52 sec

190 tasks

45 dependencies

25 plugins

Mockito

1 min 44 sec

53 tasks

14 dependencies

20 plugins

Ratpack

6 min 37 sec

594 tasks

278 dependencies

38 plugins

vy

29 sec

asks

pendencies

ugins

Grails

9 min 31 sec

567 tasks

229 dependencies

30 plugins

Griffon

5 min 50 sec

1030 tasks

94 dependencies

37 plugins

SDKMAN!

44.2 sec

11 tasks

34 dependencies

13 plugins

e

43 sec

tasks

dependencies

ugins

PyGradle

1 min 1 sec

65 tasks

46 dependencies

30 plugins

Shadow

1 min 32 sec

24 tasks

15 dependencies

24 plugins

Buildship

54.1 sec

65 tasks

117 dependencies

18 plugins

WHAT'S NEXT

- First class Java 9 Support (Jigsaw)
- Distributed Cache
- Ongoing dedicated performance work
- Ongoing Buildship (Eclipse plugin) improvements

GRADLE INC

Motto: Build Happiness

Mission: To revolutionize the way software is built and shipped.

We're Hiring: Gradle is hiring front-end, back-end, and core software engineers. Visit gradle.org/jobs to apply.

THANK YOU!

- Slides and code : https://github.com/breskeby/talks/tree/master/161010-jokercon-stp
- Gradle documentation : http://gradle.org/documentation/
- Gradle Build Scans : https://gradle.com
- Sample Build Scan: https://scans.gradle.com/s/6mjjoq6hwr7kk
- Follow me: @breskeby