



Des temps de build qui cassent votre rythme ?

Découvrez comment les accélérer !

Android Makers,
Paris, Avril 2025

Quelques questions sur votre build?

- ◆ Qui a un build long ?
 - 1 heure et plus ?
 - 30 minutes et plus ?
 - 5 minutes et plus ?
 - Moins d'une minute ?
- ◆ Qui a un build complexe ?
- ◆ Qui est heureux d'utiliser un outil de build ?



Agenda

Introduction

Gradle & Android Gradle Plugin

Concepts & Définitions

Scalabilité de Gradle

Et ensuite?

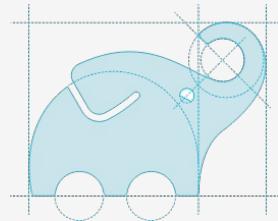


Introduction

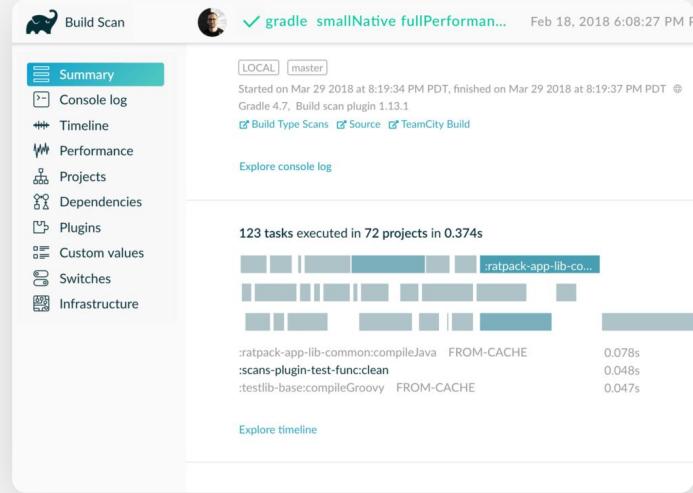




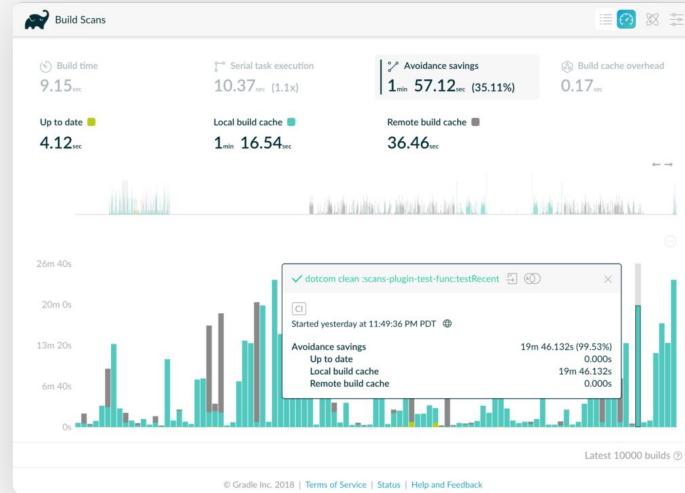
Depuis 2008, notre mission
est d'améliorer la productivité
des développeurs



Pourquoi Develocity? Pour les données.



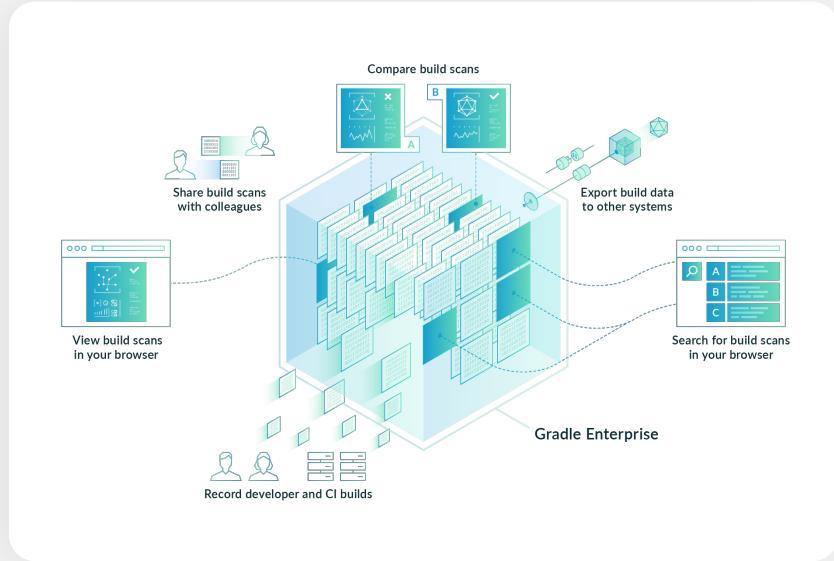
Build Scan - Donnée pour une exécution d'un build



Dashboard - Analytiques et perspectives sur les 10.000 derniers builds - en local ou sur CI



Develocity est une plateforme de données



- ◆ **Données collectives de l'équipe** - toutes les données de chaque build de tout l'équipe offre un dataset et des perspectives uniques.
- ◆ **Gestion de la Performance du Build** - vos builds restent rapides et fiables grâce aux données collectives et profondes.
- ◆ **Accélération de la résolution de problèmes** - découverte rapide des causes sources des échecs de build grâce aux données collectives et profondes.



Gradle & Android plugin



Gradle Vision

Gradle Build Tool is an open-source build system that automates the process of building software of any type, size, or complexity in a fast and reliable manner.



Gradle Vision

Gradle Build Tool is an open-source build system that automates the process of **building software of any type, size, or complexity in a fast and reliable manner.**



Gradle

01

Moteur de configuration et d'exécution.

02

APIs permettant de construire des abstractions pour les tâches communes du développement logiciel

03

Plugins permettant aux développeurs de décrire leur build, en laissant Gradle orchestrer l'exécution



Exemple d'application Java

```
plugins {
    id("application")
}

repositories {
    mavenCentral()
}

dependencies {
    testImplementation("org.junit.jupiter:junit-jupiter-engine:5.9.3")
    testRuntimeOnly("org.junit.platform:junit-platform-launcher")
    implementation("com.google.guava:guava:32.1.1-jre")
}

application {
    mainClass = "com.example.Main"
}
```



Android Gradle Plugin

01

Développé par Google, utilise les APIs de Gradle

02

Défini son modèle, ses abstractions, propres au développement Android

03

Depuis la version 7.0, le numéro de version majeur est aligné sur celui de Gradle



Exemple Android

```
plugins {  
    id("com.android.application") version "8.9.1"  
}  
  
android {  
    namespace = "com.example.app"  
    compileSdk = 34  
    // ...  
  
    defaultConfig {  
        applicationId = "com.example.app"  
        minSdk = 34  
        // ...  
    }  
}
```

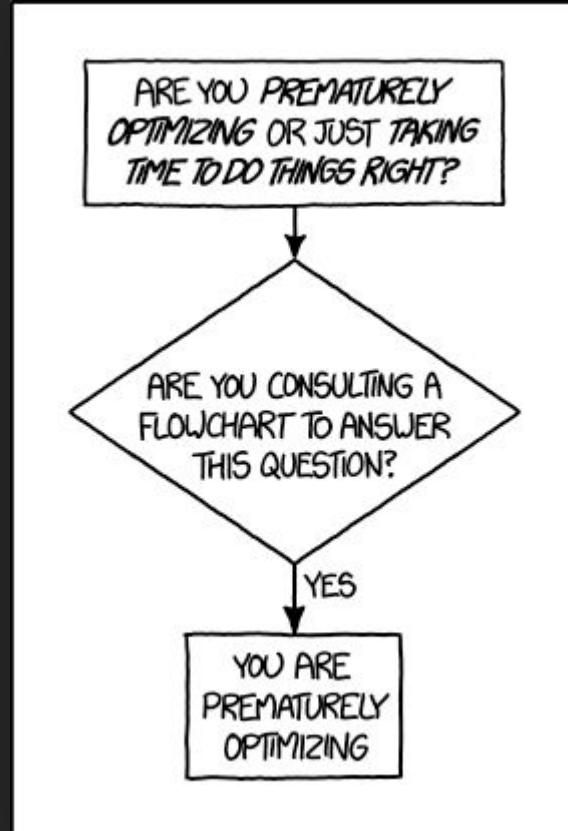


Concepts & Définitions



Mesurer c'est douter

- ◆ Mesurer
- ◆ Changer
- ◆ Douter
- ◆ Mesurer
- ◆ Comparer
- ◆ Répéter



xkcd.com/1691



Boucles de Développement

01

Boucle de développement interne
Coder, Tester, Deployer, etc...

02

Boucle de développement externe
Synchroniser l'IDE, CI persistante ou éphémère, etc...

03

Boucle de développement du build
Configurer un projet, changer la logique de build, etc...



Principes directeurs

01

L'overhead doit être proportionnel à l'impact du changement.

02

Les développeurs doivent avoir une vue compréhensible de la construction.



Approches principales

01

Évitement du travail

Ne pas refaire ce qui a déjà été fait.

02

Parallélisme

Maximisation de la vitesse en tirant parti du traitement simultané.



Scalabilité de Gradle



Scalabilité de Gradle

En pratique



Ne pas refaire ce qui a déjà été fait

Déjà disponible

	Dev	Build logic/Sync	CI	CI Éphémère
Cache de dépendances	✓	✓	✓	⚠️
Configuration cache hit	✓	✓ / N/A	✓	✗ *
Tâche à jour?	✓	✓	✓	✗
Evitement de compilation (ABI)	✓	✓	✓	✗



Ne pas refaire ce qui a déjà été fait

Déjà disponible

	Dev	Build logic/Sync	CI	CI Éphémère
Build cache local	✓	✓	✓	
Build cache distant (Develocity)	✓	✓	✓	✓



Ne pas refaire ce qui a déjà été fait

Déjà disponible

	Dev	Build logic/Sync	CI	CI Éphémère
Configure on Demand (Incubating)	⚠	⚠	⚠	⚠
Tâches incrementales	✓	✓	✓	✗
Predictive Test Selection (Develocity)	✓	✓ / N/A	✓	✓



Ne pas refaire ce qui a déjà été fait

En cours de développement

- ◆ Isolated Projects
 - Configuration Cache incrémental par projet pour l'exécution de tâches
 - Cache incrémental par projet des modèles de tooling pour le sync dans Android Studio



Parallèlisme

Déjà disponible

	Dev	Build logic/Sync	CI	CI Éphémère
Chargement CC en parallèle (8.11)	✓	✓ / N/A	✓	?
Ecriture CC en parallèle (Incubating 8.11) i	✓	✓ / N/A	✓	?
Parallelisme dans le project via CC	✓	✓ / N/A	✓	?
Parallelisme entre projets via <code>--parallel</code> i	✓	✓	✓	✓



Parallélisme

Déjà disponible

	Dev	Build logic/Sync	CI	CI Éphémère
Téléchargements en parallèle	✓	✓	✓	✓
Exécution parallèle dans une tâche (Worker API)	✓	✓	✓	✓
Test Distribution (Develocity)	✓	✓ / N/A	✓	✓
Construction des modèles de tooling en parallèle pour Android Studio	N/A	N/A / ✓	N/A	N/A



Parallélisme

En cours de développement

◆ Isolated Projects

- Configuration des projets en parallèle pour l'exécution de tâches
- Configuration des projets en parallèle pour le sync Android Studio



What's next?



Forward Looking Statements Disclaimer

This document may contain forward-looking information and statements about Gradle, its subsidiaries, and their products. These statements include projections and estimates, statements regarding plans, objectives and expectations with respect to future operations, products and services, and statements regarding future performance. Forward-looking statements may be identified by the words “believe”, “expect”, “anticipate”, “target” or similar expressions. Although Gradle's management believes that the expectations reflected in such forward-looking statements are reasonable, you are cautioned that forward looking information and statements are subject to numerous risks and uncertainties, many of which are difficult to predict and generally beyond the control of Gradle, that could cause actual results and developments to differ materially and adversely from those expressed in, or implied or projected by, the forward-looking information and statements. Gradle undertakes no obligation to publicly update its forward-looking statements, whether as a result of new information, future events, or otherwise.



Gradle 9.0

RC ce mois-ci

GA début d'été

- ◆ Kotlin 2.1 & Language Version 2.1
 - Nullabilité plus stricte avec JSpecify
- ◆ Groovy 4
- ◆ Java ≥ 17 pour exécuter Gradle 
 - Worker, compilation et tests acceptent Java ≥ 8
- ◆ Configuration cache: Plus d'erreur en utilisant les plugins core de Gradle
 - Mais certains désactivent la configuration cache car incompatible
- ◆ Et 2 ans et plus de nettoyage d'APIs 



Gradle 9.x et après

Ne pas refaire ce qui a déjà été fait

- ◆ Dépréciation de la désactivation du Configuration Cache
- ◆ Cache de build de nouvelle génération
- ◆ Cache de plus haut niveau (sous graphe de tâches)
- ◆ Worker daemons réutilisables
- ◆ Focus



Gradle 9.x et après

Parallélisme

- ◆ Exécution de tâches en streaming (via CC)
- ◆ Améliorations globales de la contention (réduction des verrous)



Public Gradle Build Tool Roadmap

<https://roadmap.gradle.org>



Merci !



```
speaker {  
    name = "Louis Jacomet"  
    twitter = "@ljacomet"  
    mastodon = "@ljacomet@foojay.social"  
    github = "ljacomet"  
}
```



```
speaker {  
    name = "Paul Merlin"  
    mastodon = "@eskatos@mastodon.social"  
    github = "eskatos"  
}
```



Questions ?

