Don’t Let Ephemeral CI Kill Your Developer Productivity

Louis Jacomet - Gradle

Gradle

Table of Contents

- Problem statement
- Gradle build profile
- Possible actions

Problem statement

- Gradle Build Tool performance

vs.

- Ephemeral environments trend

Gradle Build Tool performance

- Enable parallel execution
- Enable the Gradle daemon
- Enable the configuration cache
- Enable incremental build for custom tasks
- Enable the build cache
- Create build for specific developer workflows
- Increase the heap size
- Optimize configuration
- Optimize dependency resolution
- Optimize [Java|Android] projects

Ephemeral environments

- Industry trend
- Isolation → no state problems
- Short lived → no clean up jobs

Gradle Build Tool performance

- Enable parallel execution ✅
- Enable the Gradle daemon ❓
- Enable the configuration cache ❓
- Enable incremental build for custom tasks ❌
- Enable the build cache ✅
- Create build for specific developer workflows ✅
- Increase the heap size ✅
- Optimize configuration ✅
- Optimize dependency resolution ❌
- Optimize [Java|Android] projects ❓

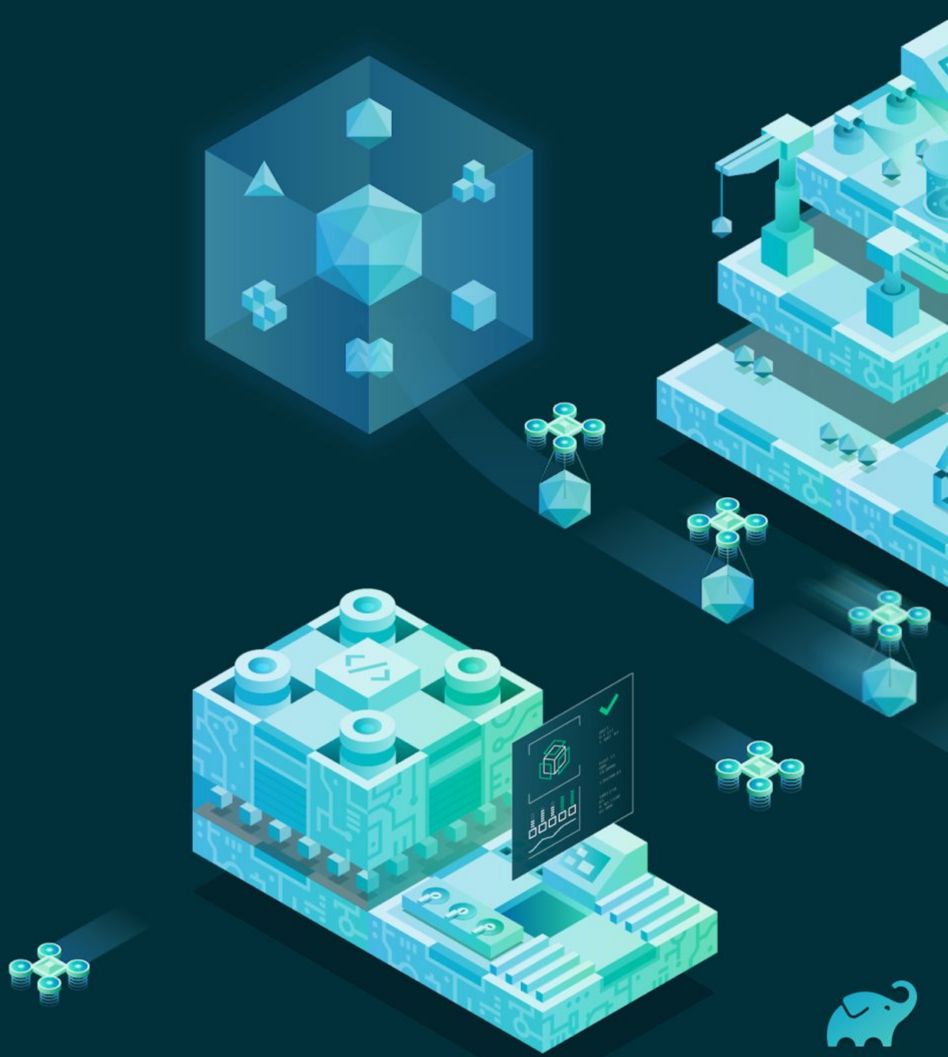
Key Gradle Build Tool performance elements

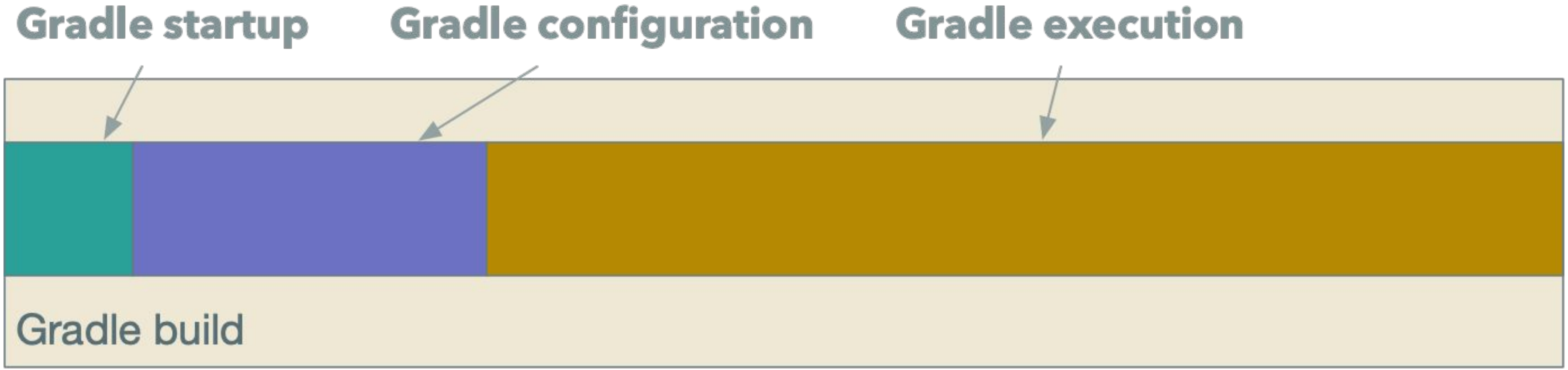
- Caches
  - Dependency cache
  - Task cache
  - …
- Incrementality
  - Execution history
- Parallelism
  - Tasks
  - Tests

Key Gradle Build Tool performance elements

- Caches ❓
  - Dependency cache
  - Task cache
  - …
- Incrementality ❌
  - Execution history
- Parallelism ✅
  - Tasks
  - Tests

Gradle Build profile





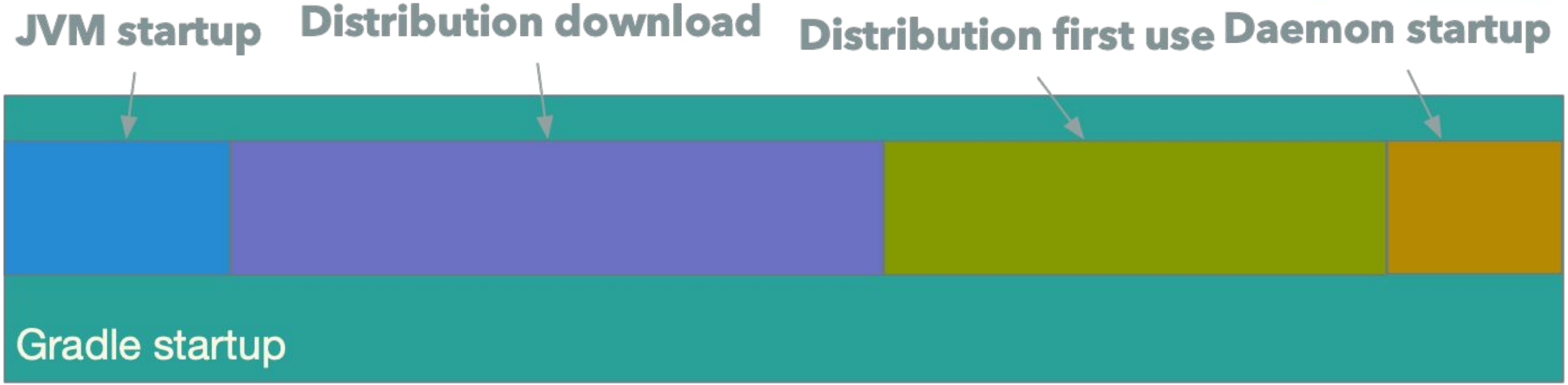
Gradle startup

Gradle configuration

Gradle execution

Gradle build

Bars not at scale



JVM startup

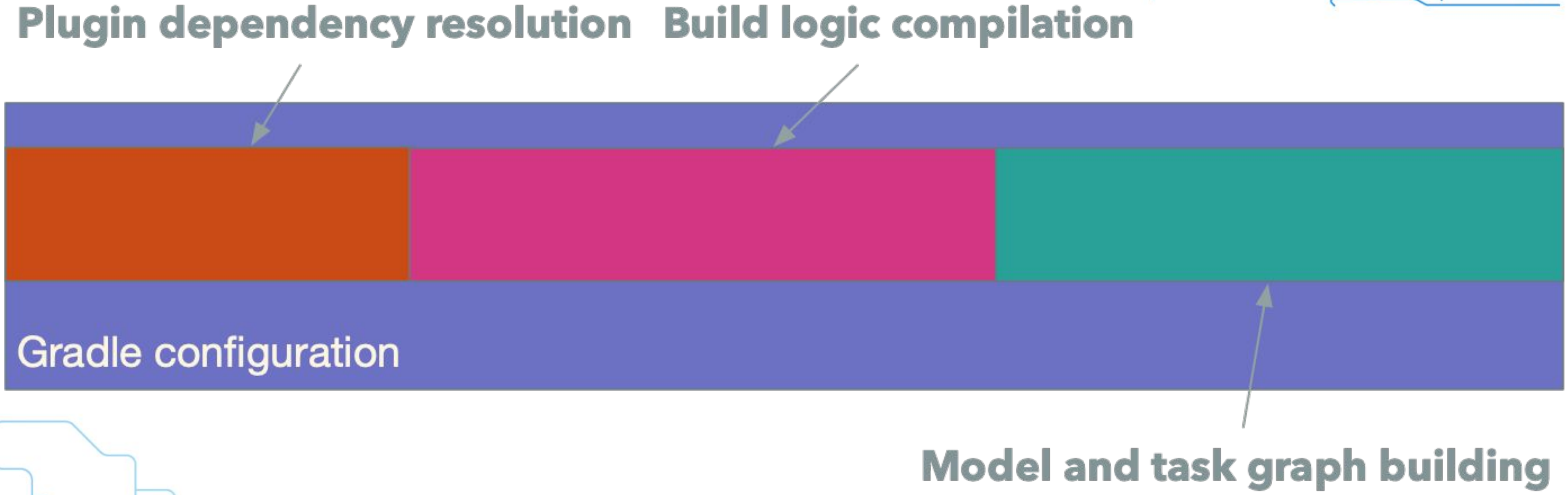
Distribution download

Distribution first use

Daemon startup

Gradle startup

Bars not at scale



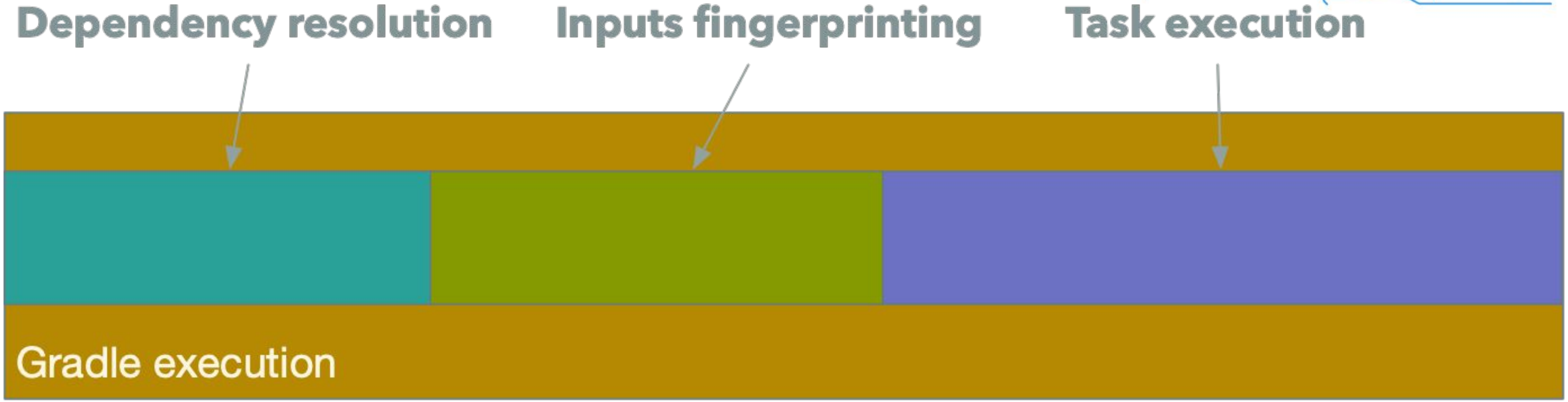
Plugin dependency resolution

Build logic compilation

Gradle configuration

Model and task graph building

Bars not at scale



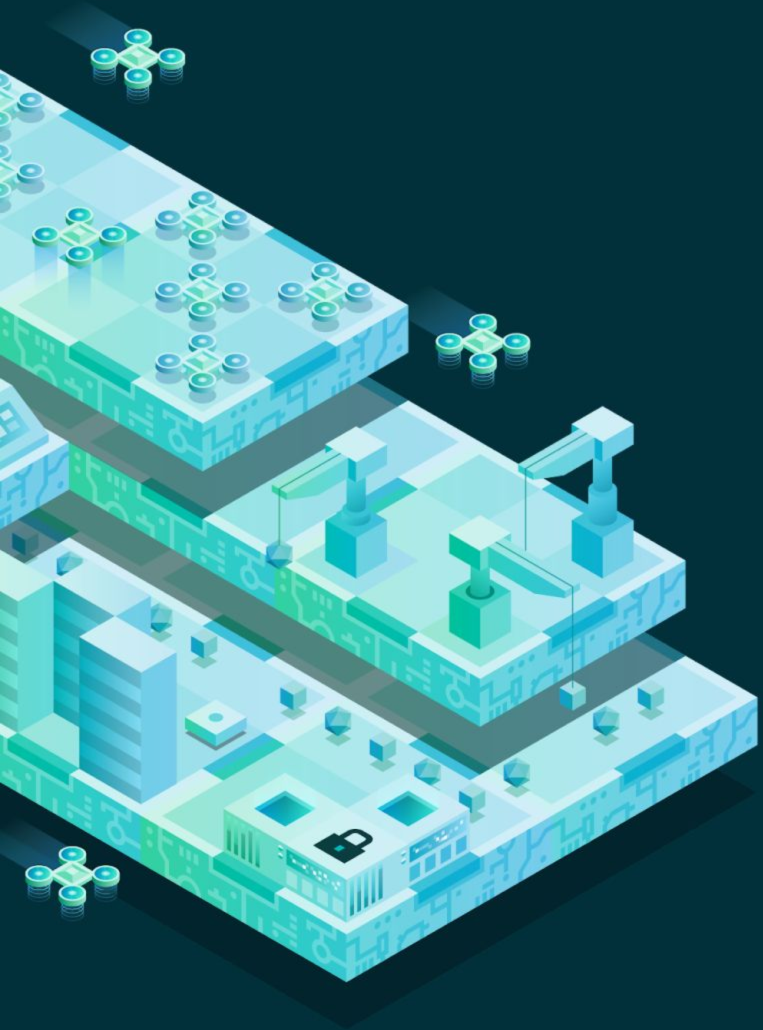
Dependency resolution

Inputs fingerprinting

Task execution

Gradle execution

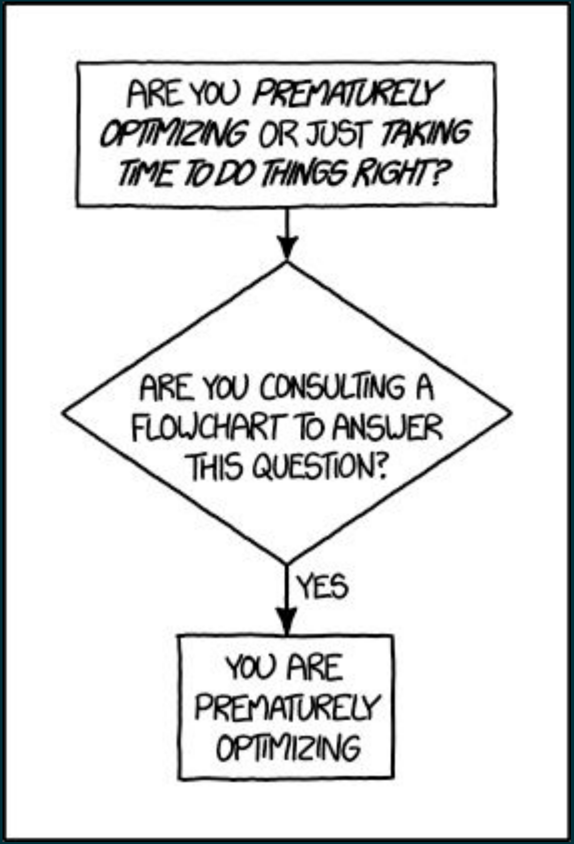
Bars not at scale



Possible actions

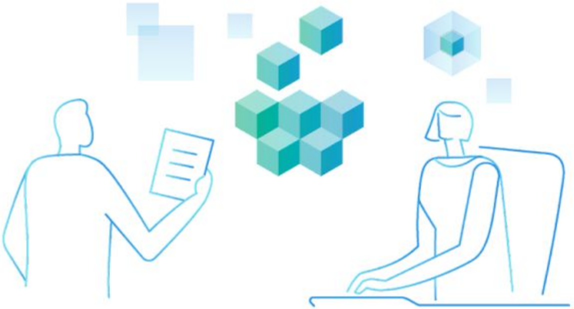
Dealing with performance

- Measure
- Change
- Measure
- Compare

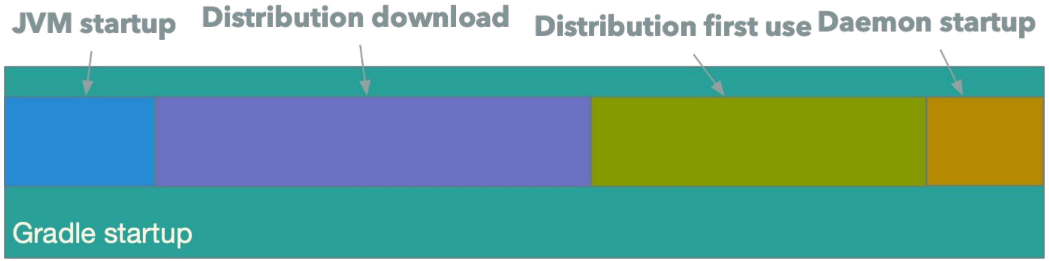

ARE YOU PREMATURELY OPTIMIZING OR JUST TAKING TIME TO DO THINGS RIGHT?
ARE YOU CONSULTING A FLOWCHART TO ANSWER THIS QUESTION?
YES
YOU ARE PREMATURELY OPTIMIZING

https://xkcd.com/1691/

Optimize Gradle startup

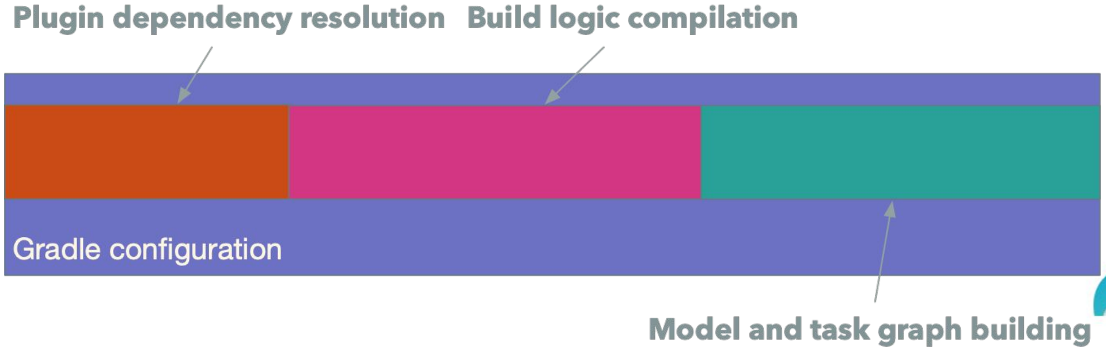


- Distribution availability
  - Always use the `-bin` one
  - Already in the image / docker file / …
  - OR Downloaded from a closer location
  - OR Save and restore `<GUH>/wrapper/dists`
- Prime distribution
  - Run it once to have the first use elements
  - OR Save and restore `<GUH>/caches/<version>/generated-gradle-jars`



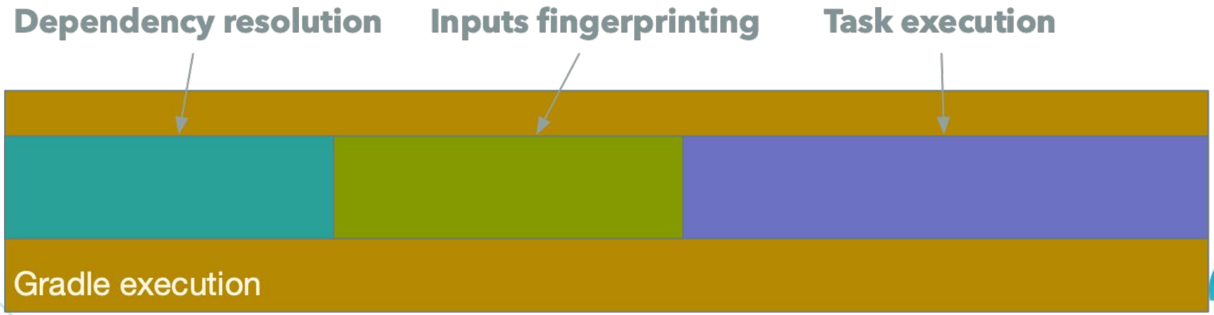
Optimize Gradle configuration

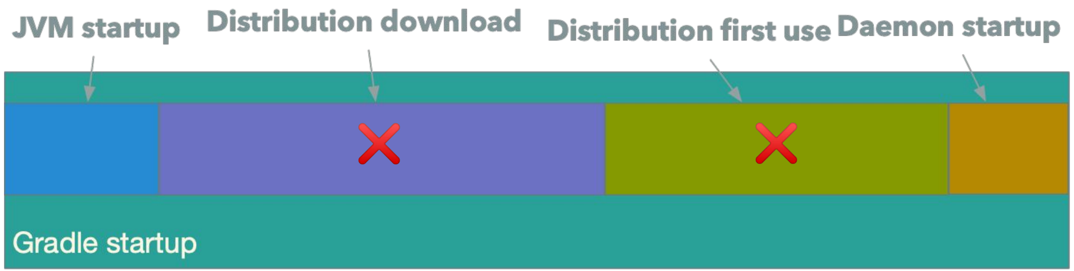
- Dependency cache
  - Read-only cache feature
  - OR save and restore `<GUH>/caches/modules-2`
- Script compilation cache
  - Remote build cache
  - OR save and restore `<GUH>/caches/<version>/kotlin-dsl` and `<GUH>/caches/<version>/scripts` and `<GUH>/caches/jars-9`



Optimize Gradle execution

- *Dependency cache*
- Task execution cache
  - ○ Remote build cache
  - ○ OR save and restore `<GUH>/caches/build-cache-1`
  - ○ (Android mostly) save and restore `<GUH>/caches/transforms-3`





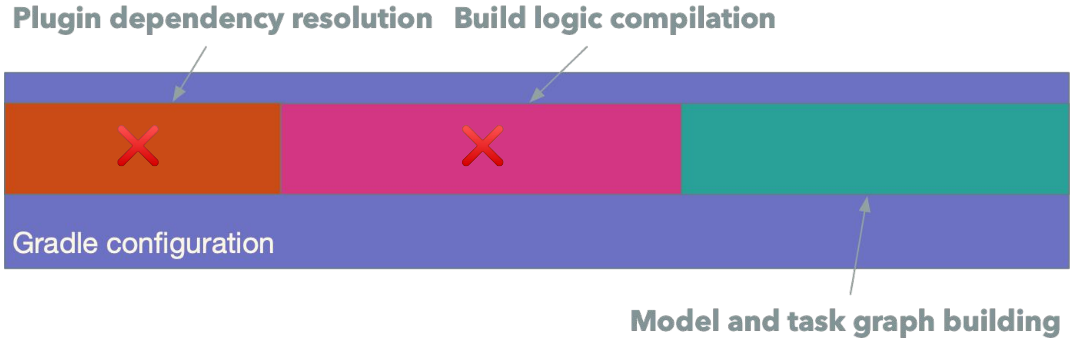
JVM startup

Distribution download

Distribution first use

Daemon startup

Gradle startup

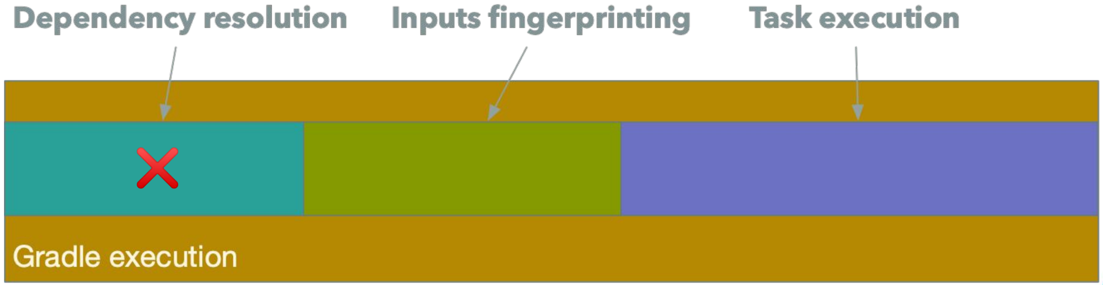


Plugin dependency resolution

Build logic compilation

Gradle configuration

Model and task graph building



Dependency resolution

Inputs fingerprinting

Task execution

Gradle execution

Thank you!