

Gradle on Steroids

Sumit Das @6amedev

Chirag Aggarwal @chi6rag

Disclaimer

Your Mileage May Vary

Always Measure First

Gradle Build Lifecycle Phases

Initialization



Configuration



Execution

Initialization and Configuration

The overhead tax you have to pay whenever you
invoke gradle

Measuring the overhead

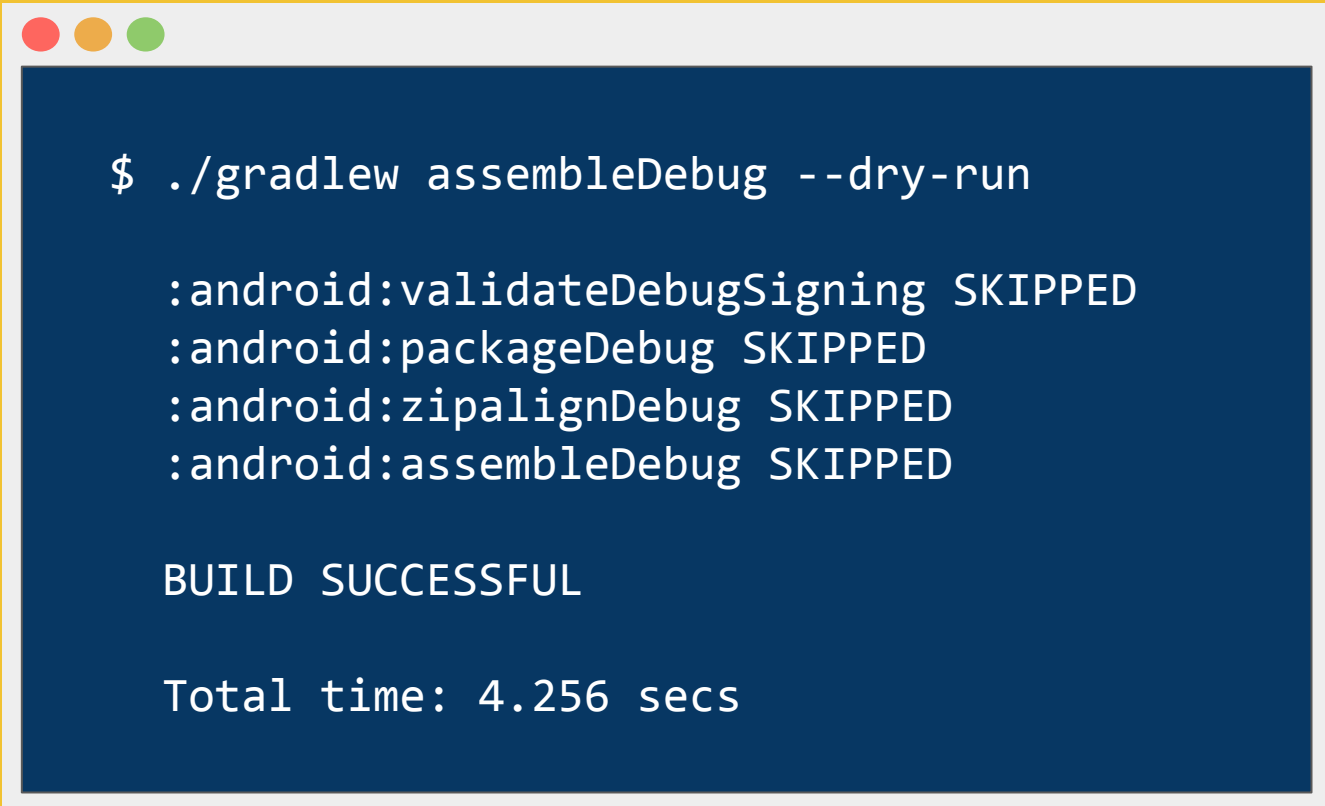
Test Subject: Google IO App

- 28 libraries
- 53149 method references

<https://github.com/google/iosched>

Measuring the Overhead

The `--dry-run` command line flag skips running all the tasks



```
$ ./gradlew assembleDebug --dry-run

:android:validateDebugSigning SKIPPED
:android:packageDebug SKIPPED
:android:zipalignDebug SKIPPED
:android:assembleDebug SKIPPED

BUILD SUCCESSFUL

Total time: 4.256 secs
```

Configuration on Demand



Android




android



server

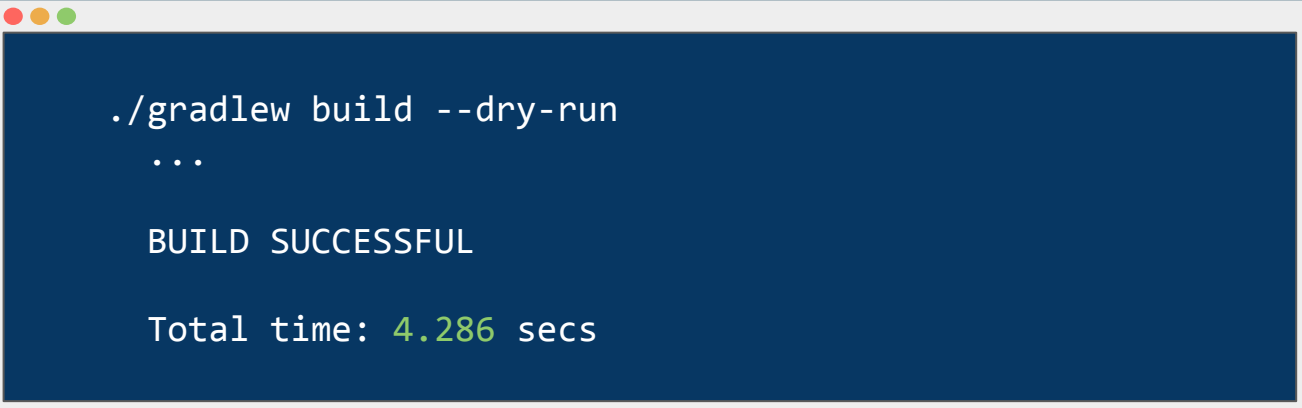


Gradle Scripts



```
./gradlew assembleDebug -dry-run  
--configure-on-demand
```

Without --configure-on-demand



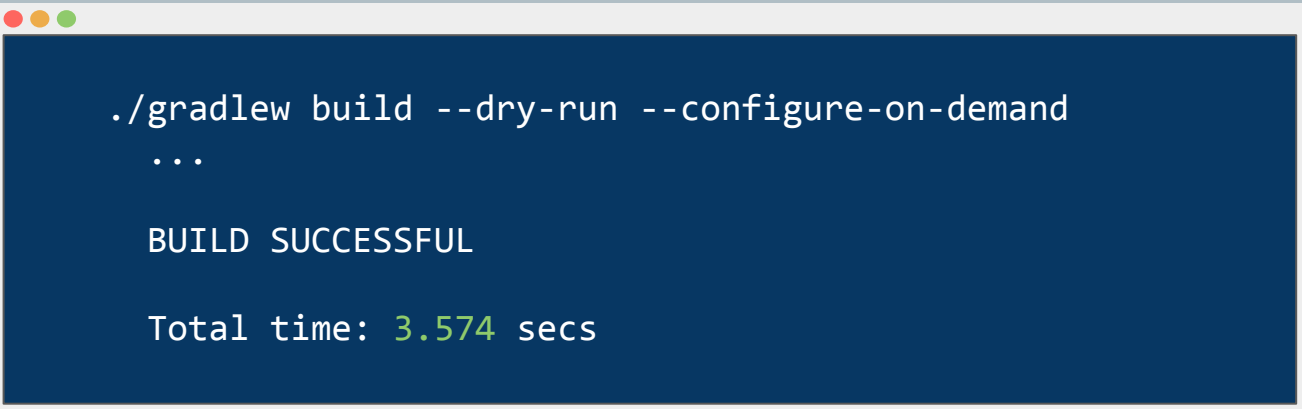
```
./gradlew build --dry-run
...

BUILD SUCCESSFUL

Total time: 4.286 secs
```

A terminal window with a dark blue background and white text. It shows the output of a Gradle build command. The text is: `./gradlew build --dry-run`, followed by `...`, then `BUILD SUCCESSFUL`, and finally `Total time: 4.286 secs`. The window has a title bar with three colored circles (red, yellow, green) on the left.

With --configure-on-demand



```
./gradlew build --dry-run --configure-on-demand
...

BUILD SUCCESSFUL

Total time: 3.574 secs
```

A terminal window with a dark blue background and white text. It shows the output of a Gradle build command with the `--configure-on-demand` flag. The text is: `./gradlew build --dry-run --configure-on-demand`, followed by `...`, then `BUILD SUCCESSFUL`, and finally `Total time: 3.574 secs`. The window has a title bar with three colored circles (red, yellow, green) on the left.

Enable Configure on Demand in Preferences > Build, Execution, Deployment > Compiler

The screenshot shows the 'Preferences' dialog box with the 'Compiler' section selected in the left sidebar. The 'Configure on demand' checkbox is checked and highlighted with a red box.

Build, Execution, Deployment

- Build Tools
- Compiler**
- Coverage
- Debugger
- Instant Run
- Required Plugins
- Languages & Frameworks**
- Tools**
- Other Settings**

Command-line Options:

Example: --stacktrace --debug (for more information, please read Gradle's [documentation](#).)

☐ Make project automatically (only works while not running / debugging)

☐ Use in-process build

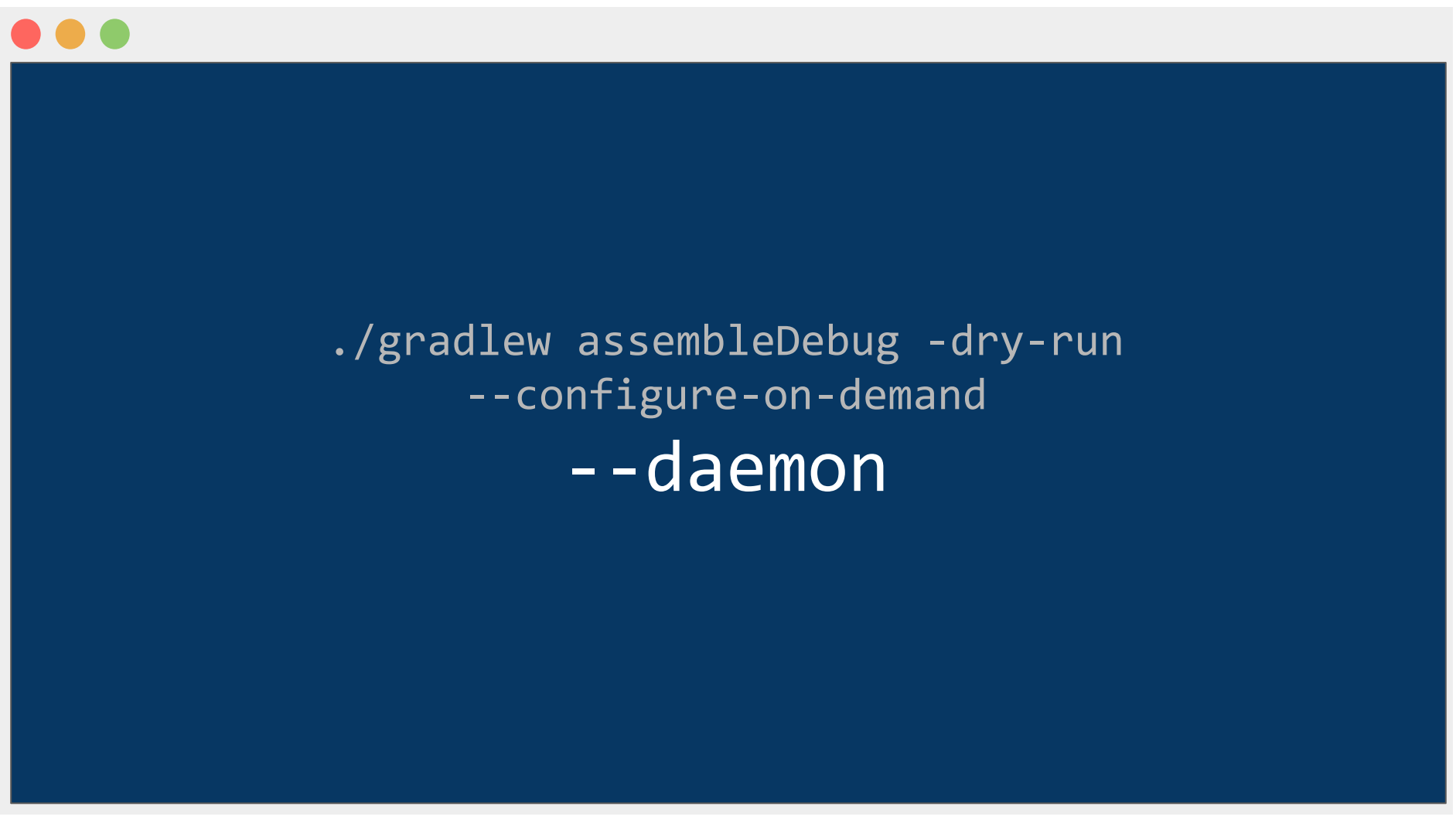
Faster, and integrated with the "Gradle Console."

☒ **Configure on demand**

This option may speed up builds. This option is in "incubation." Please read Gradle's [documentation](#).

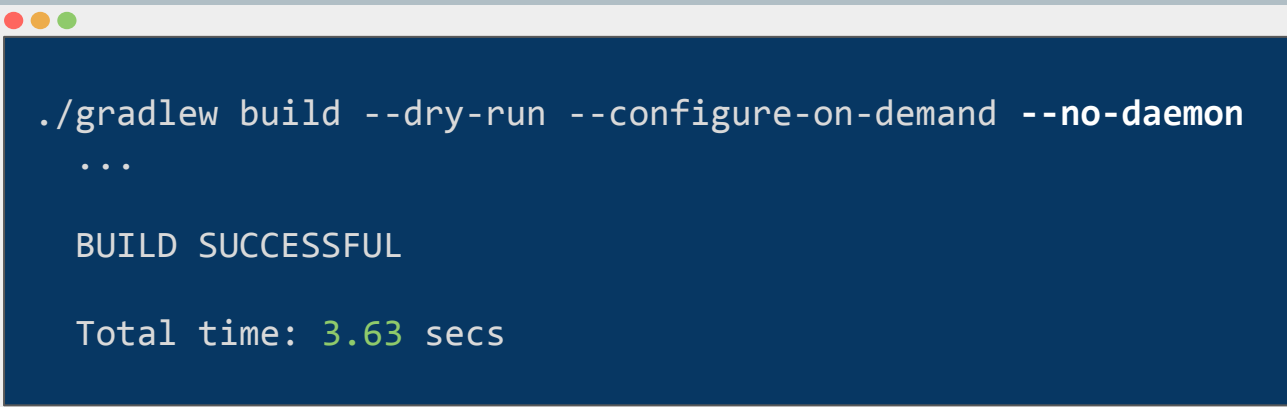
Enable the Daemon

- Keeps JVM instance running in background
- Speeds up startup time



```
./gradlew assembleDebug -dry-run  
--configure-on-demand  
--daemon
```

Without --daemon

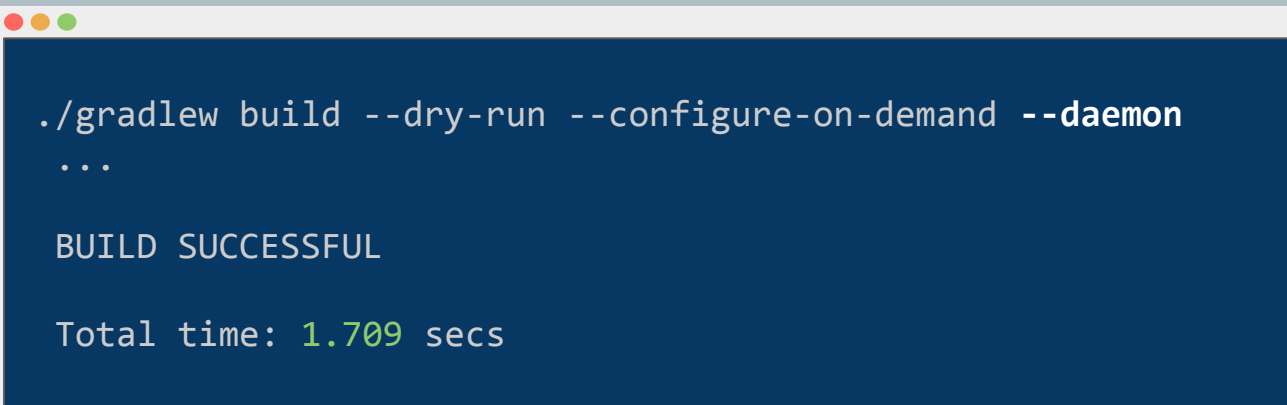
A terminal window with a dark blue background and white text. The window has three colored window control buttons (red, yellow, green) in the top-left corner. The text inside the terminal shows the command to run a Gradle build without the daemon, followed by an ellipsis, the success message, and the total time taken.

```
./gradlew build --dry-run --configure-on-demand --no-daemon
...

BUILD SUCCESSFUL

Total time: 3.63 secs
```

With --daemon

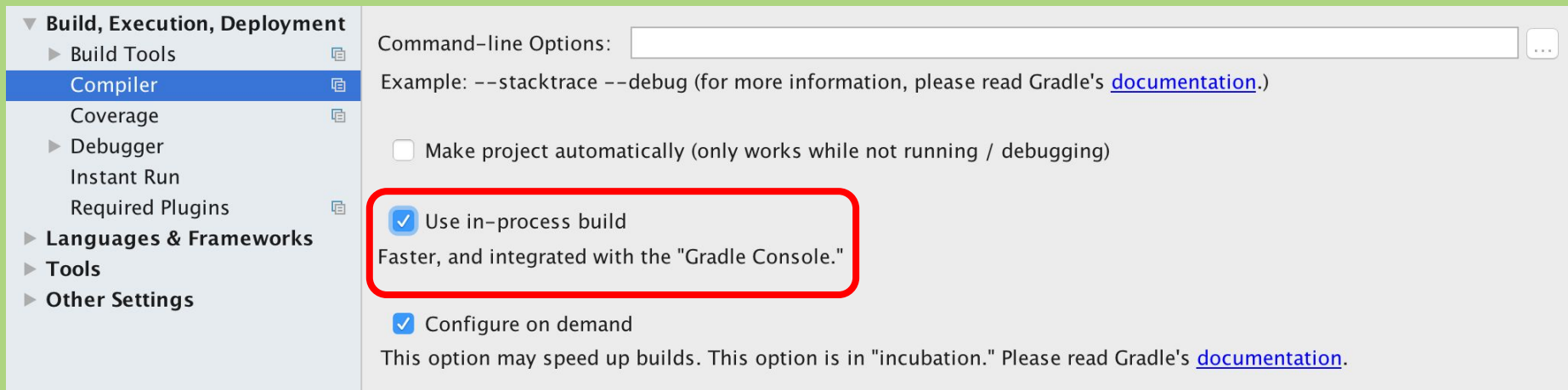
A terminal window with a dark blue background and white text. The window has three colored window control buttons (red, yellow, green) in the top-left corner. The text inside the terminal shows the command to run a Gradle build with the daemon, followed by an ellipsis, the success message, and the total time taken.

```
./gradlew build --dry-run --configure-on-demand --daemon
...

BUILD SUCCESSFUL

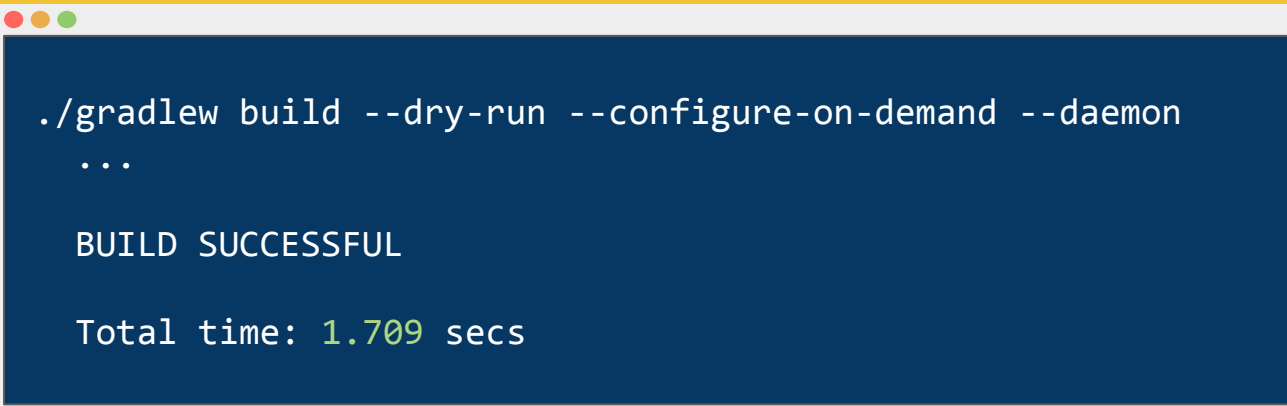
Total time: 1.709 secs
```

Enable Configure on Demand in Preferences > Build, Execution, Deployment > Compiler



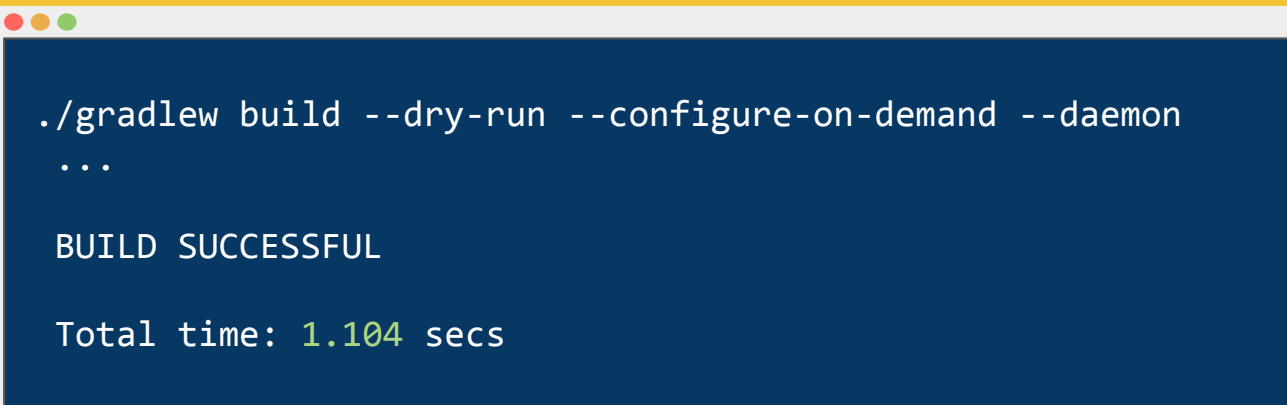
Use latest Gradle
Distribution

Gradle 2.2.1

A terminal window with a dark blue background and white text. The window has a title bar with three colored dots (red, yellow, green) on the left. The text inside the terminal shows the execution of a Gradle build command with various flags, followed by an ellipsis, the message 'BUILD SUCCESSFUL', and the total time taken.

```
./gradlew build --dry-run --configure-on-demand --daemon  
...  
  
BUILD SUCCESSFUL  
  
Total time: 1.709 secs
```

Gradle 2.14

A terminal window with a dark blue background and white text. The window has a title bar with three colored dots (red, yellow, green) on the left. The text inside the terminal shows the execution of a Gradle build command with various flags, followed by an ellipsis, the message 'BUILD SUCCESSFUL', and the total time taken.

```
./gradlew build --dry-run --configure-on-demand --daemon  
...  
  
BUILD SUCCESSFUL  
  
Total time: 1.104 secs
```

Gradle Version

Upgrade Gradle Version in
`gradle/wrapper/gradle-wrapper.properties`




```
#Tue Dec 02 14:11:25 EST 2014
distributionBase=GRADLE_USER_HOME
distributionPath=wrapper/dists
zipStoreBase=GRADLE_USER_HOME
zipStorePath=wrapper/dists
distributionUrl=https\://services.gradle.org/distributions/gradle-2.14-all.zip
```



Issues with using dynamic versions

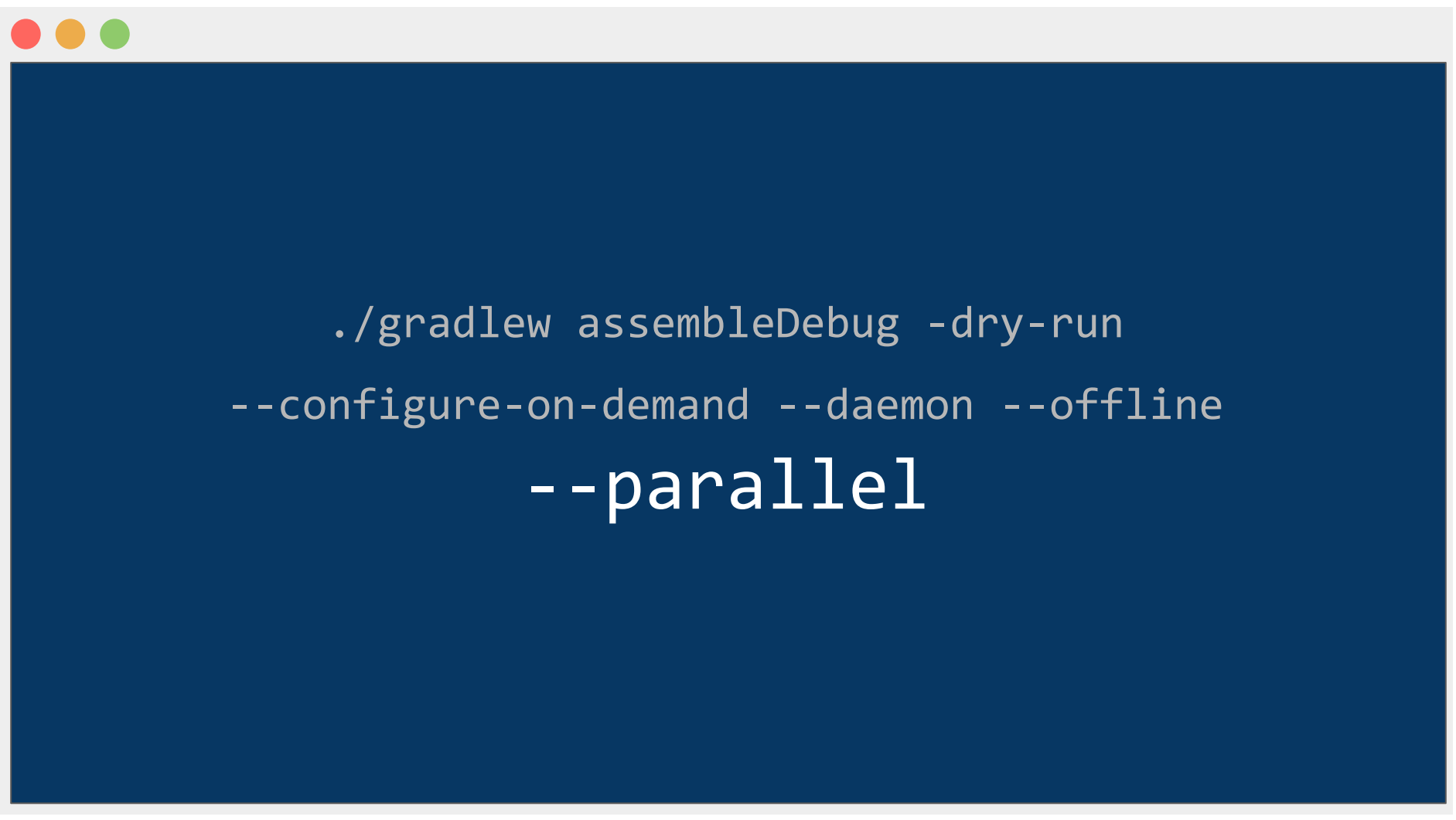
```
dependencies {  
    compile 'com.google.code.gson:gson:2.+'  
}
```

Offline Mode



```
./gradlew assembleDebug -dry-run  
--configure-on-demand --daemon  
--offline
```

Parallel Builds

A terminal window with a dark blue background and a light gray title bar. The title bar contains three colored window control buttons (red, yellow, green) on the left. The terminal text is displayed in a light gray monospaced font, with the last line in white.

```
./gradlew assembleDebug -dry-run  
--configure-on-demand --daemon --offline  
--parallel
```


...but when is it useful?

- Multi-project build
- Decoupled projects
- No single project dominates build time

Apply Plugins
Judiciously

Apply Plugins Judiciously

Android Maps Utils : Polyline Decoding

`com.google.maps.android:android-maps-utils:0.4.4`

Methods count: **1228**

Apply Plugins Judiciously

Google Play Services : Maps

Don't use

`com.google.android.gms:play-services:9.4.0`


Methods count : **74646**

Use

`com.google.android.gms:play-services-maps:9.4.0`

Methods count : **15306**

Avoid Expensive
or
Blocking Tasks



```
def cmd = 'git rev-list HEAD --count'
def gitVersion = cmd.execute().text.trim().toInteger()

android {
    defaultConfig {
        versionCode gitVersion
    }
}
```

Splitting `build.gradle`

CI Gradle File

Developer Gradle File

Dev Gradle



Building all flavours and variants.



Generating test coverage reports.

CI Gradle



Dev Gradle



Compiling libraries:
Crashlytics, New Relic, etc.



Expensive/blocking tasks

CI Gradle



But what happens when you add a dependency to Dev gradle file and forget to add it in CI gradle file?

Build Fails

Sharing common dependencies
between

Dev and CI gradle files

`dependencies.gradle`

Centralising Libraries

`libraries.gradle`

Centralising Libraries

- In multi-module projects, it is very effective in restricting usage to a single version of a given library.
- It makes maintaining and upgrading libraries a lot easier.

Grouping libraries by vendor
for compatibility

```
ext {  
    version = '23.4.0'  
    supportLibraries = [  
        cardView      : "cas:cardview-v7:${version}",  
        recyclerView: "cas:recyclerview-v7:${version}",  
        design         : "cas:design:${version}"  
    ]  
}
```

- `com.android.support` is replaced with `cas`

Removing unused dependencies

Other Performance Tweaks

Boost daemon's heap size

Invest in fast CPUs

Other Performance Tweaks

Boost daemon's heap size

Invest in fast CPUs

Thank you.