Build time regression when upgrading gradle plugin from 4.1.2 to 4.2.0-beta06 or 7.0.0-alpha12

Infeasible Bug P2 + Add Hotlist

STATUS UPDATE No update yet. Edit

DESCRIPTION de...@gmail.com created issue #1

IMPORTANT: Please read
https://developer.android.com/studio/report-bugs.html carefully and supply
all required information.

Studio Build: Android Studio Arctic Fox 2020.3.1 Canary 9
Version of Gradle Plugin: 7.0.0-alpha11
Version of Gradle: 6.8.2/6.8.3
Version of Java: 11
OS: Debian

Steps to Reproduce:
1. Building with gradle plugin com.android.tools.build:gradle:4.1.2 takes approx. 25 mins
2. Building with com.android.tools.build:gradle:7.0.0-alpha12 takes approx. 1h10mins

The only change is updating the version of the gradle build tools.

Using the gradle profile we see that signing time consumption goes from 4s to 45 mins.

Attached are outputs of
gradlew -Pandroid.enableProfileJson=true
gradlew --profile
For the slow builds.

gradle-profile.zip
20 KB Download

android-enableProfileJson.zip
38 KB Download

✓ Links (3)

Links (3)

"https://developer.android.com/studio/report-bugs.html carefully and supply"
" ...for the information, I was able to find a similar bug filed to openjdk here ."
" ..., 16,384 native calls are made that use the 8MB input array and the 512 byte output array. These arrays are accessed using GetPrimitiveArrayCritical that with -Xcheck:jni copies the array: http://hg.

COMMENTS

dn...@google.com <dn...@google.com>

Assigned to an...@google.com.

je...@google.com <je...@google.com>

Reassigned to sp...@google.com.

sp...@google.com <sp...@google.com> #2

Reassigned to am...@google.com.

Amr, it's the `FinalizeBundleTask` that's taking ~40 minutes. Could you take a look?

am...@google.com <am...@google.com> #3

Thanks for filing the bug, could you describe the resources you have in your project? whether you have so many small resources or not many but very large in size? Also what is the heap size

de...@gmail.com <de...@gmail.com> #4

In our main project, we have many small primarily in XML.
res folder size: 680K

Number of resources: 149
Number of png: 10 (the biggest is approx. 15k)

We load a native sdk as a AAR.

Size of the jni (*.so files): 16.9MB spread on 14 files (for arm64-v8a)

Size of assets: 8.8MB mix of .mkv, .wav, .jpg, and .png (the biggest is approx. 440KB).

We found that we had
-Xcheck:jni in org.gradle.jvmargs
In the project gradle file. Removing that drastically reduced the execution time. 1h to 10 mins on 7.0.0-alpha11. We haven't tested 4.1.2 with this configuration.

Please let us know if you think it is a misconfiguration.

---

am...@google.com <am...@google.com> #5

Status: Won't Fix (Infeasible)

Thanks for the information, I was able to find a similar bug filed to openjdk here.

> To compress the array, 16,384 native calls are made that use the 8MB input array and the 512 byte output array. These arrays are accessed using GetPrimitiveArrayCritical that with -Xcheck

That explains why the `FinalizeBundleTask` became orders of magnitude slower, as starting 4.2, the task is responsible for both signing and compressing the bundle.

I'd suggest using the option only when encountering JNI problems and not by default.

I'm closing this bug and filing a bug internally for potentially warning when the option is used.