Android studio debug causes app to run/execute at a very slow speed on initial ADB install, and after resume from an inline break-point

Bug P3 + Add Hotlist

STATUS UPDATE No update yet. Edit

DESCRIPTION cm...@gmail.com created issue #1 Sep 23, 2022 1

Build: AI-212.5712.43.2112.8815526, 202207101541,

AI-212.5712.43.2112.8815526, JRE 11.0.12+0-b1504.28-7817840x64 JetBrains s.r.o., OS Linux(amd64) v5.15.0-48-generic, screens 3840.0x2160.0, 2560.0x1600.0

AS: Chipmunk | 2021.2.1 Patch 2; Kotlin plugin: 212-1.7.10-release-333-AS5457.46; Android Gradle Plugin: 4.2.2; Gradle: 6.9; Gradle JDK: version 11.0.12; NDK: from local.properties: (not specifie
SDK: 22.1.7171670; LLDB: pinned revision 3.1 not found, latest from SDK: (package not found); CMake: from local.properties: (not specified), latest from SDK: 3.18.1-g262b901, from PATH: 3.22.

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

About 2 days ago, for unexplained reason after a fresh adb installation, the app debug execution speed is running at an extremely slow pace; it even causes aTalk login process to timeout on a 3(
timer; this was never happened before.
This slow execution speed persists throughout the whole debug process. When the USB debug cable is detached (no adb debug process is attached) , everything returns to normal operation.

In the past the exact same problem happens only after the app resumes from an inline break-point.
Whenever this happens, I just restart a whole new adb debug process. However the above new problem has now prevented one to do any form of useful app debug; except to dump/view the logc

In both case, the slow app execution speed is very serious, and practically prevents anyone to perform any useful app debug.

idea.zip
344 KB Download

aTalk_timeout.jpg
54 KB View Download

✓ Mentioned issues (1) ✓ Links (3)

Mentioned issues (1)

P1 ByteBuffer works extremely slow when debugger is attached "Possible dup of b/264292265" aa...@ #5, cm...@

Links (3)

"IMPORTANT: Please read https://developer.android.com/studio/report-bugs.html carefully and supply all required information."
"Actually the whole aTalk library is available on github, https://github.com/cmeng-git/atalk-android ."
"To see the problem, you can download https://github.com/cmeng-git/hymnchtv source to perform the debug, the whole app will be running at a very slow pace once any break point is hit." cm...@ #

COMMENTS All comments

jv...@google.com <jv...@google.com> Sep 24, 20

Assigned to an...@google.com.

ju...@google.com <ju...@google.com> Sep 24, 20

Status: New

em...@google.com <em...@google.com> #2 Sep 26, 2022 05

First, is it possible that you have one of these in Android Studio?

1. Java/Kotlin method breakpoint
2. Java/Kotlin field watchpoint
3. C/C++ watchpoint
4. C/C++ conditional breakpoint

Can you open the "Breakpoints" panel in both debuggers (Java and Native) and verify the contents?

Second, when this happens, can you open the Android Studio native debugger's LLDB tab, and then type:

```
breakpoint list
```

...and share the result here?

cm...@gmail.com <cm...@gmail.com> #3 Sep 27, 2022 10

> Second, when this happens, can you open the Android Studio native debugger's LLDB tab, and then type:
I am unable to issue this comment on LLDB window; a notification pop up i.e. "This field is read-only". So I added the "breakpoint list" in [Edit Configuration | Debugger | LLSB Post Attach Con
an item

Followings are the observed results. Each test is carried upon restart of the app debug mode.
Java method breakpoint
: No breakpoints currently set.

Java field watchpoint
No breakpoints currently set.

Java line breadpoint
: No breakpoints currently set.

... verify the contents?
Not sure what you are looking for, so I did not capture the value.
So I just provide a screenshot for the line breakpoint

I am unable to perform the following, as most of the jni are external pre-built as static .so libraries.
C/C++ watchpoint
C/C++ conditional breakpoint

Actually the whole aTalk library is available on github, https://github.com/cmeng-git/atalk-android.
Your team can download and carry out more detail investigation.

By the way, the slow execution speed problem upon new adb installation just vanish for unexplained reason, 2 days after I raised this bug.
I am not sure if this will come back again.

LineBreakPoint.png
245 KB View Download

aa...@google.com <aa...@google.com> #4 Aug 15, 2023 04

What device is this on? Specifically, what API level?

aa...@google.com <aa...@google.com> #5 Aug 15, 2023 04

Possible dup of b/264292265

cm...@gmail.com <cm...@gmail.com> #6 Aug 15, 2023 09

Note: although the slow execution upon first adb installation does not occur any more on all my test devices.

However below is still happen on all the devices API tested i.e. API-21, API-24, API-28, API-29, API-33.

In the past the exact same problem happens only after the app resumes from an inline break-point. Whenever this happens, I just restart a whole new adb debug process. However the above problem has now prevented one to do any form of useful app debug; except to dump/view the logcat message.

aa...@google.com <aa...@google.com> #7 Aug 15, 2023 10

I don't understand the last comment.

What is still happening? What is not happening?

cm...@gmail.com <cm...@gmail.com> #8 Aug 15, 2023 10

I am rephrased my previous statement:

1. Note: although the slow execution upon first adb installation does not occur any more on all my test devices.
This refers to the issue referred to: Possible dup of [b/264292265](https://issuetracker.google.com/issues/264292265)

i.e. the problem described in my first comment..
"About 2 days ago, for unexplained reason after a fresh adb installation, the app debug execution speed is running at an extremely slow pace; it even causes aTalk login process to timeout or
seconds timer; this was never happened before.
This slow execution speed persists throughout the whole debug process. When the USB debug cable is detached (no adb debug process is attached) , everything returns to normal operation.

2. However below problem is still happening on all the devices API tested i.e. API-21, API-24, API-28, API-29, API-33.

In the past the "exact same problem" happens only after the app resumes from an inline break-point.
Whenever this happens, I just restart a whole new adb debug session. However this slow execution problem has prevented one to do any form of useful app debug; except to dump/view the l
message.

The "exact same problem" refers to slow debug execution.

aa...@google.com <aa...@google.com> #9 Aug 16, 2023 12

Are you saying that it takes a long time to resume from a breakpoint?

Can you attach a video?

---

cm...@gmail.com <cm...@gmail.com> #10 Aug 16, 2023 09

No, what I meant is that after the app is resumed from a break point. The whole app execution is running at a very slow pace, possible more than 100x slower prior to any break point hit. After this, it is not further practical to continue with the debug at this slow execution speed. I need to restart the app to return the app execution back to its normal pace.

To see the problem, you can download https://github.com/cmeng-git/hymnchtv source to perform the debug, the whole app will be running at a very slow pace once any break point is hit.

---

aa...@google.com <aa...@google.com> #11 Aug 16, 2023 10

What devices are you using? Is this happening with the built-in emulators in Android Studio?

Can you confirm that it's happening on all API levels?

We are aware that is happening on AI 33 and used to happen on API 32 until it was recently patched with a fix. But in my testing, I could not reproduce this with ny other API levels.

---

cm...@gmail.com <cm...@gmail.com> #12 Aug 16, 2023 10

Previously the problem happened on all the API's that I have tested, API-21, API-26, API-31, API-33, for both AVD and real devices.

With your latest input, I need to retest them to see if the problem still exist.
I have not doing any system debug for the past 2-3 months. By the way, when the problem was since resolved.

---

aa...@google.com <aa...@google.com> #13 Aug 17, 2023 07

I built and installed https://github.com/cmeng-git/hymnchtv and added:

```
    private void onNumberClick(View btnView) {
        Random random = new Random(System.currentTimeMillis());
        long now = System.currentTimeMillis();
        for (int i = 0; i < 1_000_000; i++) {
            Math.sin(random.nextInt());
        }
        Toast.makeText(this, "Time: " + (System.currentTimeMillis() - now), Toast.LENGTH_SHORT).show();
        ...
```

I don't see any slowdown when debugging with or without a breakpoint on API 31 or 32. I do see a significant slowdown on API 33 with or without a breakpoint (b/264292265).

---

cm...@gmail.com <cm...@gmail.com> #14 Aug 17, 2023 08

I have just conducted a quick debug test on the following devices:
a. Samsung Note10+ device
b. AVD Pixel2 API-31
c. AVD Pixel 6 Pro API-33

The slow execution speed after break point do not happen on all the 3 devices. Look like the problem has been resolved.
You can close the issue for now. I will continue monitoring in future testing..

Thanks a lot for your team effort to fix the issue.