IssueTracker   Search IssueTracker   Sign in

Android Public Tracker   172946944

Native library linked list searching performance slowdown in android 11.

+1 7   Hotlists (13)   Mark as Duplicate

Comments (13)   Dependencies   Duplicates (1)   Blocking (0)   Resources (3)

Assigned   Bug   P3   +   [AOSP] assigned

STATUS UPDATE No update yet. Edit

DESCRIPTION ky...@gmail.com created issue #1   Nov 11, 2020 01:58PM

The search speed of the linked list used in the native library on my pixel 2 was more than 2x slower on Android 11 compared to Android 10.

I can't guess what caused this to happen, so I ask for help here.

It has 450,000 nodes, and the code is as follows.

```
struct Node {
   char data[33];
   Node *next;
}


bool list_search(char *data) {
      Node temp = NULL;
      temp = Node_Header;

      while (NULL != temp) {
            if (strcmp(data, temp->data) == 0) {
                  return true;
            }

            temp = temp->next;
      }

      return false;
}
```

Links (3)   Hide all

"For steps to capture a bug report, please refer: https://developer.android.com/studio/debug/bug-report#bugreportdevice"	am...@ #2
" https://drive.google.com/drive/folders/1d7qA8opxQFMdPG-kWjreaK-vTDPm... "	ky...@ #3
" ..., i'll have a look later this week. in the meantime, you might find https://developer.android.com/studio/profile/android-profiler usef... "	en...@ #11

COMMENTS   All comments   Oldest first

am...@google.com <am...@google.com> #2   Nov 11, 2020 07:05PM

Assigned to am...@google.com.

Thank you for the report.

*Can you please let us know how this behavior is tested?

*Did you use any sample apk to test the behavior, if so, please share the apk and share the steps that we can use to replicate the issue and how to make out the wrong behavior from the apk

*Please share the bugreport too.

Android bug report (to be captured after reproducing the issue)
For steps to capture a bug report, please refer: https://developer.android.com/studio/debug/bug-report#bugreportdevice

Alternate method
Navigate to "Developer options", ensure "USB debugging" is enabled, then enable "Bug report shortcut". Capture bug report by holding the power button and selecting the "Take bug report" option.

Note: Please upload the files to google drive and share the folder to android-bugreport@google.com, then share the link here.

ky...@gmail.com <ky...@gmail.com> #3   Nov 12, 2020 05:24PM

* Bug Report And Sample files.
https://drive.google.com/drive/folders/1d7qA8opxQFMdPG-kWjreaK-vTDPm7Fhc?usp=sharing

Reporter	ky...@gmail.com
Type	Bug
Priority	P3
Severity	S3
Status	Assigned
Access	Default access View
Assignee	en...@google.com
Verifier	--
Collaborators	
CC	an...@google.com ky...@gmail.com ma...@google.com
AOSP ID	--
ReportedBy	Developer
Found In	--
Targeted To	--
Verified In	--
In Prod	

* How to run sample app
  Just press "LINKEDLIST" button.
  then your will see result on the screen with time.

I shared screenshot too.
  Pixel2_Android11.jpg
  Nexus5_Android8.1.jpg

  Pixel2
   - OS : Android 11
     - Build : RP1A.201005.004

  Nexus 5X
   - OS : Android 8.1.0
     - Build : OPM7.181205.001

This screenshot is the result of running the test app, and you can see that the Pixel2 took longer than the Nexus 5X.

I previously tested on Android 9, 10 and 11 versions with a Pixel2 and got the slowest results on Android 11.

** sorry for my bad english

Thank you.

am...@google.com <am...@google.com> #4 Nov 13, 2020 01:31AM

Thank you for reporting this issue. We've shared this with our product and engineering teams and will continue to provide updates as more information becomes available.

ma...@google.com <ma...@google.com> #5 Nov 13, 2020 05:54AM

A quick question: Does the regression scale with the number of nodes in the search, i.e. it gets slower compared to Android 10 the more nodes you add?

ky...@gmail.com <ky...@gmail.com> #6 Nov 13, 2020 12:31PM

Here is my test data

Device : Pixel 2
Android 10 Build : QQ3A.200805.001

Node Count	R	Q
100,000	11ms	1ms
200,000	21ms	2ms
300,000	32ms	3ms
400,000	45ms	4ms
500,000	53ms	5ms
600,000	66ms	6ms
700,000	73ms	6ms
800,000	87ms	7ms
900,000	101ms	8ms
1,000,000	109ms	9ms

ma...@google.com <ma...@google.com> #7 Nov 14, 2020 10:36AM

Thank you, that helps narrow down where the problem may be. Just to be clear, the app isn't reloaded within each test, right?

ky...@gmail.com <ky...@gmail.com> #8 Nov 16, 2020 11:21AM

I changed the app slightly so that I could set the number of iterations.
and each test, I deleted it from the recent list and ran the app again.

en...@google.com <en...@google.com> #9 Nov 17, 2020 05:36AM

Reassigned to en...@google.com.

the link only includes a bug report. please provide the sample source code so we can try to reproduce this.

ky...@gmail.com <ky...@gmail.com> #10 Nov 17, 2020 11:13AM

The 172946944_BugReport.zip file in Google Drive contains the test app's project.

en...@google.com <en...@google.com> #11 Nov 17, 2020 11:51AM

okay, i'll have a look later this week. in the meantime, you might find https://developer.android.com/studio/profile/android-profiler useful.

en...@google.com <en...@google.com> #12

Nov 20, 2020 11:09AM

i can reproduce R being slower than P or Q (on pixel 3 xl, since i don't have a 2) using your apk. i couldn't work out how to rebuild your JNI code (rather than use the prebuilt .so you have) so i created a new project based on the "hello-jni" sample and copied the "active ingredients" from your code into that project:

```
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        val binding = ActivityHelloJniBinding.inflate(layoutInflater)
        setContentView(binding.root)

        setup();
        for (i in 0..10) {
            val t0 = System.currentTimeMillis()
            search();
            val t1 = System.currentTimeMillis()
            Log.i("###","took: " + (t1 - t0) + "ms")
        }
    }

    external fun setup()
    external fun search()
```

and

```

std::vector<char*> TestVector;

extern "C" JNIEXPORT void JNICALL Java_com_example_hellojni_HelloJni_setup(JNIEnv* env, jobject thiz) {
    for (int i = 0; i < 457760; ++i) {
        TestVector.push_back(strdup("ABCDEFGHIJABCDEFGHIJABCDEFGHIJAA"));
    }
}

extern "C" JNIEXPORT void JNICALL Java_com_example_hellojni_HelloJni_search(JNIEnv* env, jobject thiz) {
    const char *data = "AAAAAAAAAABBBBBBBBBBCCCCCCCCCCAA";

    for (std::vector<char*>::iterator iter = TestVector.begin(); iter != TestVector.end(); iter++) {
        if (strcmp(data, *iter) == 0) {
            return;
        }
    }
}
```

which is the core of your original. (i deliberately switched to the hard-coded string to take the cost of ART's GetStringUTFChars() out of the equation.)

the "good" news is that i *can't* reproduce this on S. but it does look like there was something wrong in R. profiling shows me strcmp_default in R and __strcmp_aarch64 in S, but they seem to be the same code (and pulling libc.so off my R and S devices seems to confirm that).

ky...@gmail.com <ky...@gmail.com> #13

Nov 20, 2020 05:46PM

I built native library with "app/src/main> ndk-build -C jni" command

Is there any way to improve performance in R?