Analyzing the Performance of C/C++ and Debugging OpenGL ES* Frames on Mainstream x86 and ARM* Android* Devices

Introduction
When developing an Android* application, you usually need to test, optimize, and debug on many different platforms. While basically every hardware and chip manufacturer provides a set of custom tools for this, developers can find it cumbersome to install and learn how to use all of them. With the latest releases (2014 R2 as of now), Intel® Graphics Performance Analyzers (Intel® GPA), Intel® Frame Debugger, and Intel® VTune™ Amplifier are able to support most mainstream Android 4.x devices regardless of their architecture (ARM* or x86).

We have tested these Intel tools and successfully used them on the ARM Android devices listed below, but basically any Android 4.x device is probably compatible. (Note: ARM*-based devices not shown in the following list are unsupported but may still work with the Intel GPA toolset.):

<table>
<thead>
<tr>
<th>Model</th>
<th>GPU</th>
<th>Android* Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Samsung* Galaxy Nexus (GT-i9250)</td>
<td>Imagination Technologies* PowerVR SGX540</td>
<td>Android 4.3</td>
</tr>
<tr>
<td>Samsung* Galaxy S4 (GT-I9500ZNASER)</td>
<td>Imagination Technologies* PowerVR SGX544</td>
<td>Android 4.4</td>
</tr>
<tr>
<td>Samsung* Galaxy S III (GT-i9300)</td>
<td>ARM* Mali 400MP</td>
<td>Android 4.3</td>
</tr>
<tr>
<td>Google* Nexus 4</td>
<td>Qualcomm* Adreno 320</td>
<td>Android 4.4</td>
</tr>
<tr>
<td>Google* Nexus 5</td>
<td>Qualcomm* Adreno 330</td>
<td>Android 4.4</td>
</tr>
<tr>
<td>LG* G2 D802</td>
<td>Qualcomm* Adreno 330</td>
<td>Android 4.4</td>
</tr>
<tr>
<td>Amazon* Kindle Fire HD</td>
<td>Imagination Technologies* PowerVR SGX544</td>
<td>Amazon* Fire OS 3.0</td>
</tr>
</tbody>
</table>

Intel® GPA System Analyzer – High-Level View of App Performance
Intel GPA System Analyzer can run on Windows* 7/8.x, Mac* OS X*, and Linux* and can be used to analyze apps running on almost any Android 4.x device connected over ADB. It’s a free tool, part of the Intel GPA suite you can get with dedicated tools for graphics analysis and optimization, or INDE, the native cross-platform productivity suite supporting Intel® Architecture and ARM*. If your device is rooted, this tool can analyze any application installed
on the device. Otherwise, it can only analyze apps with Internet permission and the debuggable flag set to “true”.

Setting the debuggable flag to “true” in the AndroidManifest is done automatically when you're doing a debug build with gradle or ant / Android Studio or Eclipse*. But when it comes to analyzing the real performance of an application, you want to run a version that has been optimized for release.

You can manually set this flag to “true” in your manifest, but lint will complain about it and cause your build to abort. You can easily overcome that.

```plaintext
android {
    lintOptions {
        disable 'HardcodedDebugMode'
    }
}
```
Now, when running, you first get a list of applications running on the device you've chosen:
Selecting one will trigger its launch on the device, along with its real-time analysis:

From there, you can drag-and-drop any metric from the left side to see its value evolve in real time. You may not get hardware-specific metrics on every device as their support varies, but you'll have at least: Target App CPU Load, RAM usage, Device and Network IOs, OpenGL* metrics (draw calls, vertex count, frame time).
In addition to these real-time values, you have the ability to trigger "state overrides" from the bottom left of the window. This will help you determine the current bottleneck in your application:

Normal scene:  With all the textures set to a simple 2x2 one:

With disabled alpha-blending:  Showing wireframe:

You can also disable all the Draw calls. Then, no graphics will be rendered, but if the FPS improves, the bottleneck isn't on the OpenGL* side.

A more in-depth article on Intel GPA along with a sample code is available: https://software.intel.com/en-us/android/articles/using-the-intel-gpa-system-analyzer-to-improve-android-apps

**Intel® Frame Debugger – Debugging an OpenGL Rendered View**

Intel Frame Debugger is used to capture and fully debug OpenGL frames. It's also a free tool you can get from https://software.intel.com/en-us/vcsource/tools/intel-gpa or INDE and fully works on most Android devices.

As of now, it can be run on Windows 7/8.x platforms only.
First, add a frame from a device by clicking "Add".
This will show you a list of applications that you may, or may not, have the right to analyze. Just like with Intel GPA System Analyzer, you may analyze any application on a rooted device. Otherwise, you can analyze only the ones with Internet permission and the debuggable flags set to “true”: 
Double-click to launch one and click on "capture" to get a frame. Once it's done, go back to the previous view and select the captured frame:

You can see all the draw calls that have been made on the left side and the associated simulated OpenGL ES frame buffer on the right side.
A double click on one of these draw calls will allow you to inspect all steps of the OpenGL ES pipeline associated with the draw call, among these:

- The primitives and the associated Vertex Buffer Objects

- The vertex and fragment shaders:

You can see and modify these shaders and even modify the values they’re receiving. This will modify the simulated frame as well as the rendered frame on your device if it's still connected!
Intel® VTune™ Amplifier – Analyzing the Performance of C/C++ Code from Android Applications

Intel VTune Amplifier is part of Intel® System Studio suite. Unlike the previously mentioned tools, this suite isn’t free unless it’s for non-commercial software development. As of now, this tool can be run from Windows and Linux hosts.

To begin with VTune Amplifier, create a new project:
In its Properties, choose your device and “Launch Android Package” as type, then you can enter your package name or click on “browse” to look for it:
Once this is done, you can launch a "basic hotspot" analysis. Starting the analysis will launch the app on your device:

Click on “Stop” to stop the collection (and automatically close the application).

The trace will be collected and displayed:
From there you can start looking at the results with the “Bottom-up” view and group data by Module / Function / Call Stack:

To associate full library symbols and sources with your results, you can add their paths to your project properties (usually ./obj/local/ABI/ for the non-stripped libs and ./jni for the sources).

More information on performance analysis using VTune can be found at: https://software.intel.com/en-us/node/471808

On supported x86 Android Devices (Intel® reference designs and Dell Venue* 8 with developer image), VTune Amplifier can be used to analyze Java* workloads, get CPU and GPU metrics, and do power analysis. You can find more details at the following links:

Notices

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS OTHERWISE AGREED IN WRITING BY INTEL, THE INTEL PRODUCTS ARE NOT DESIGNED NOR INTENDED FOR ANY APPLICATION IN WHICH THE FAILURE OF THE INTEL PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an order number and are referenced in this document or other Intel literature may be obtained by calling 1-800-548-4725, or going to:
http://www.intel.com/design/literature.htm

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark* and MobileMark*, are measured using specific computer systems, components, software, operations, and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products.

Any software source code reprinted in this document is furnished under a software license and may only be used or copied in accordance with the terms of that license.

Intel, the Intel logo, and VTune are trademarks of Intel Corporation in the U.S. and/or other countries.

Copyright © 2014 Intel Corporation. All rights reserved.

*Other names and brands may be claimed as the property of others.