Power and Performance Optimization on Android*

In this article I give a brief overview of power and performance optimization methodology on mobile devices. I hope it will help you better understand the challenges involved. I also provide instructions for how to measure device power consumption from the battery controller on Android.

Why is power and performance optimization so challenging on mobile devices? Let’s quickly review how power and performance optimization is performed in HPC (High Performance Computing).

The important metrics regarding power consumption in HPC are:

- Maximum power consumption - to calculate power plant and cooling equipment
- Performance per watt – to measure computational effectiveness
- Idle power consumption – to reduce power usage when no applications are running

Most of the optimizations for these metrics are done at the hardware or operating system level. Applications just need to process workloads as fast as possible; so it’s more about performance optimization. The first step is to implement the best algorithm and data structures for the task. The second step is to maximize the usage of all the available resources (nodes, network, accelerators, CPUs, cores, etc.) to reduce the time it takes to process a workload. When the workload is finished, the system switches to the idle state with minimal power consumption.

Power optimization on mobile devices is more complicated:

- Mobile devices contain a lot of hardware components
- Mobile devices typically don’t run computational workloads

The most important power metric for mobile devices from the user’s perspective is battery life. With a given battery capacity this metric directly depends on the device power consumption. So we will use device power consumption as the key metric.

Mobile devices have a lot of hardware components

Mobile devices have many hardware components:

- CPU
- GPU
- Wireless modules
  - Wi-Fi*
  - Bluetooth*
  - LTE / 3G / GSM
  - NFC
- Sensors
  - Gyroscope
  - Accelerometer
  - Light sensor
  - Proximity sensor
- GNSS
- Audio
- Display
Each component consumes different amounts of power depending on how applications use them. For example, Wi-Fi may be enabled or disabled; the CPU may run on several frequencies; Display Brightness may be set to different levels, etc.

Device power consumption is the sum of all power consumption used by all the components.

Mobile devices typically don’t run computational workloads

We can relatively easily measure the performance of a game application. Basic game performance analysis is reviewed in the article Guidelines on How to Run a Basic Analysis for Android* Graphical Applications Using Intel® GPA tools.

Mobile devices typically execute one or more event-based applications. Latency and responsiveness are also important. An application may be using several hardware components.

Let’s take a look at the picture below:

This device has three active applications, each using several hardware components:

- CPU
- GPU
- Wi-Fi
- LTE
- GNSS
- Accelerometer

Other hardware components are disabled or set to idle mode.

It’s very important to find ways to reduce the number of active hardware components and also reduce the power consumption of each active hardware component. You’ll use different optimization strategies for each component.

The problem is when a single application extensively uses all the available hardware components, it can make all optimization efforts useless.

It’s hard to measure power consumption per component

So we have several important metrics: the device power consumption and the power consumption of each hardware component.

Total device power consumption can be measured from the battery controller. When a device is not connected to the charger, the battery is the only power source.

It’s not typically possible to directly measure a specific component’s power usage in watts (like the Wi-Fi module) without disassembling the device and using external hardware for measurements. Of course, we could measure indirect metrics like CPU frequency, network usage, screen brightness level, etc. Intel® Graphics Performance Analyzers (Intel® GPA) can be used for these indirect power measurements.

How to measure power from battery controller

It’s possible to measure device power consumption from the battery controller using Intel GPA System Analyzer from the GUI or Intel GPA Console Client from the command line.

It’s not practical to measure power with a USB connection because most devices will charge when connected via USB. It’s possible to switch ADB daemon on the device to TCP/IP mode and connect via a
Wi-Fi network. The computer with the Android SDK and the Android device should be connected to the same Wi-Fi network.

Connecting the device with ADB over TCP/IP:

1. Connect the device to host with USB cable.
2. Run: “adb tcpip 5555” command on host.
3. Disconnect USB cable from the device.
4. Discover the device’s IP address in Settings.
5. Run: adb connect <device-ip> on host.
6. Check that the device is connected by running “adb devices” command.

After these steps the device is accessible using ADB but disconnected from all power sources except the internal battery. That’s why power measurements from the battery controller are the total power consumption for the device.

Collecting power metrics using Intel® GPA System Analyzer
Start Intel GPA System Analyzer and connect it to Android device. The Android device name will be the same as the one listed in “adb devices” command output.

A list of all installed applications will appear on the screen. All applications are divided into two categories: “Analyzable applications” and “Non-analyzable applications”.

Intel GPA can only analyze applications with the debuggable flag set in the application manifest on non-rooted device.

To prepare the application for analysis using Intel GPA:

1. Open the AndroidManifest.xml file
2. Add the Internet permission:
   `<uses-permission android:name="android.permission.INTERNET">
3. Set the debuggable flag to true:
   `<application android:debuggable="true">

Another option is to collect power metrics using the “System Profiling -> System View” option. No application-specific metrics (like FPS) are available in this mode, but power metrics and other system-wide metrics are available in this mode.

The available power metrics are shown in the left panel:

Drag the metrics to the center to start measurement.

Collecting power metrics using Intel® GPA Console Client
You can also obtain power metrics using Intel GPA Console Client from the command line.

To obtain the list of all supported metrics run:

$ gpa-console-client -am
Typically, these power metrics are available:

```
# Power
metric.hw.power.current_discharging  # Current Discharging
metric.hw.power.current_charging    # Current Charging
metric.hw.power.power_discharging   # Power Discharging
metric.hw.power.power_charging      # Power Charging
metric.hw.power.battery_voltage     # Battery Voltage
```

Enter the following line into `config.cfg` text file and run in “System View” mode:

```
$ gpa-console-client -an "System View" -cfg config.cfg
```

The tool will output metrics in CSV format.

To collect application-specific metrics you need to specify the application name in “-an” parameter:

```
$ gpa-console-client -an "App Name" -cfg config.cfg
```

Intel GPA can only analyze applications with the debuggable flag set in the application manifest on non-rooted device as explained above for Intel GPA System Analyzer.

The list of all installed applications can be obtained using the following command:

```
$ gpa-console-client -al
```

**Conclusion**

This article reviews power and performance optimization on mobile devices. It contains the instructions for how to measure device power consumption on Android devices from the battery controller using Intel GPA. Device power consumption is a key metric in power optimization. Successful power optimization requires developers to reduce device power consumption.

Indirect power metrics like CPU frequency or network usage can help you identify which hardware component uses a higher percentage of the device’s power.

**Related Resources and Articles**

- [Intel® Graphics Performance Analyzers](#)
- [Guidelines on How to Run a Basic Analysis for Android* Graphical Applications Using Intel® GPA tools](#)
- [Using Intel® Graphics Performance Analyzers Console Client for Android* Application Performance Analysis](#)
- [Developing Games with the Android* NDK and Intel® tools](#)

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

Copyright © 2015 Intel Corporation. All rights reserved.

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