

An Energy Abstraction Layer for Mobile Computing Devices

Mirco Josefiok
OFFIS e.V., Oldenburg
mirco.josefiok@offis.de

Marcel Schröder Andreas Winter
Carl von Ossietzky Universität, Oldenburg
marcel.s@gmx.de winter@se.uni-oldenburg.de

Abstract

Since the growing popularity of smartphones and tablet devices, energy-efficiency in mobile computing is an increasingly interesting topic. But in case of software development engineering energy-efficiency is widely neglected, even clear and simply applicable means to measure and visualize energy consumption caused by software usage is still in its infancy. This work provides basic research in the field of measuring energy and power related information on mobile computing devices and proposes an abstract specification for implementing a measurement infrastructure on different mobile computing devices.

1 Motivation

In Germany every year more than 600 Mrd. kWh of electrical energy are consumed¹. From this, more than 10% are accounted for information and communication technology [11]. Here, Energy consumption of mobile devices counts for ca. 11.6% [10], which results in half of the capacity of Germany's² most powerful nuclear power plant ISAR 2. Following the idea, that the mobile calculating power increases in the same speed, as in the past years, there is an optimization potential of about 20% to 40% of the ICT related power consumption if a holistic approach for optimizing every connected part can be found [6]. Even more, reducing energy consumption of mobile devices will also increase the lifetime of batteries due to fewer required charge cycles. But, in case of software development and engineering, improving energy-efficiency of mobile devices is widely neglected [3].

There exists many chances for optimizing energy consumption on mobile computing devices. This can be archived on different levels, ranging from hardware, operating system, and machine code to application level [4]. Low-level software optimization and improving machine code are adequately studied [8]. Optimizing energy consumption from a software point of view may vary from using alternative algorithms, resource substitution, applying compression techniques purposefully, using user profiles for energy efficient process scheduling, applying alternative hard/software-sensors, identifying and eliminating energy code smells etc. [1].

Validating these approaches and showing their con-

tribution on saving energy requires sufficiently grained measurement techniques. Offline measurement, by using external measurement devices, is not always possible, since this requires breaking mobile devices. Online measurement, by using software means provided by the operation system, often does not furnish sufficient precision and relies on certain details of the operating system capabilities. This work aims at providing an *Energy Abstraction Layers* (EAL) in the field of mobile computing. For this purpose the EAL will abstract measurement capabilities and provide unified access to them independently from the device it is used on.

2 Requirements for an EAL

In this section the requirements for an abstract measurement specification will be described. Functional requirements resemble the intended functionality of the the EAL in an implemented form on a concrete platform. The following (most central) functional and non-functional requirements have been conducted [5]:

The EAL must provide power and energy measurement functionality. Per device and platform exists various ways of gathering energy and power related information (e.g. battery capacity, electricity and power consumption etc.). Access of those information is even restricted to certain access levels or only available on some devices.

The EAL must provide measurement functionality for estimating the runtime of each hardware component. Different hardware components in different states of operation require different energy consumption. Knowing their runtime, and combining these data with appropriate energy consumption models leads to meaningful information on hardware related energy consumption.

The EAL has to respect and report accuracy. Different measurement techniques result in different accuracy. It is the desired to know the accuracy of a measurement method [12].

The EAL should provide measurement capabilities per application and component. This would be especially helpful for finding energy wasting code patterns in applications [2].

The EAL should provide suitable error handling. As not every device and platform supports every measurement method, the EAL should report, if certain measurement capabilities are not present.

The EAL must be platform independent. For mobile devices, there exist at least two major platforms (Android, iOS) which have to be treated analogously.

¹http://www.ag-energiebilanzen.de/component/download.php?filedata=1326461230.pdf&filename=BRD_Stromerzeugung1990-2011%2020Dez2011&mimetype=application/pdf

²<http://www.kernenergie.de/kernenergie/themen/kernkraftwerke/kernkraftwerke-in-deutschland.php>

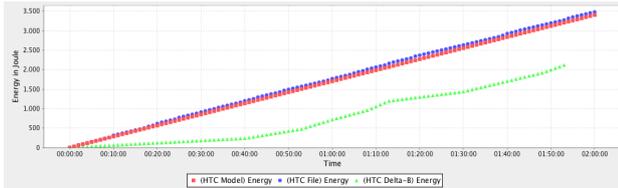


Figure 1: Energy consumption measured with three different methods on a HTC One X.

The EAL must be in device independent. Different devices offer different possibilities for measuring power consumption. Furthermore, energy efficiency of applications also depends on used hardware components. *The EAL should not rely on hardware measurement procedure.* Measuring energy consumption may rely on various online techniques (cf. [7]).

3 Implementation

Most of the demanded requirements of the EAL were implemented in a prototype to measure the energy consumption on Android devices [9]. Measuring energy consumption requires to access voltage and the discharging current. While the voltage could be easily read by Android API methods, the discharging current is more complicated to access. The current EAL implementation provides three different techniques to estimate the discharging current. Figure 1 shows the energy consumption measured by these online techniques for running a simple GPS application.

The first and simplest technique is Delta-B measuring. Delta-B compares the changes of the battery charge level in relation to the battery capacity. All needed values are derived by using the `BatteryManager` class of Android API which is available since API level 5. Since this method is based on the battery charge level, Delta-B only provides 100 real measurement values, which finally only approximates for a realistic trend.

The second method reads the discharging current from the file system (File). Providing these files relies on the devices vendors, so these undocumented system files are not available on every device and updated in different time intervals. On a HTC One X the discharging current was updated every 60 seconds which is more accurate than the Delta-B method.

The third technique relies on energy profiles (Model) provided by the devices vendor. The average power consumption of different hardware components, like display in different brightness levels, enabled states for GPS, Bluetooth, CPU speed etc. are made available in an XML-file of the Android device. Android internally collects the up-time of these hardware components. A hidden part of the Android API provides methods to access these data. Comparing data measured at different times allows to calculate the enabled time and the related energy consumption in a millisecond resolution for many hardware components. This technique strongly depends on the quality of the provided energy model. Using this model

based technique also allows to calculate the distribution of energy consumption related to different hardware components.

4 Summary

This paper shortly introduced the need for a standardized technique for measuring the energy consumption of apps running on mobile devices. Based on a set of requirements, an implementation of such an API was presented, which provides three different measurement techniques resulting in different accuracy. This API was already applied in [2] so show a trend of effectivity for energy smell refactorings. But further experiments on applying these techniques to measure energy consumption are required to show their reliability and to validate their benefit.

References

- [1] C. Bunse, S. Naumann, A. Winter. Entwicklung und Klassifikation energiebewusster und energieeffizienter Software. In J. Marx Gómez, C. V. Lang, V. Wohlgemuth, (eds.), *IT-gestütztes Ressourcen- und Energiemanagement, Konferenz zu den 5. BUIS-Tagen*, to appear. Springer, 2013.
- [2] M. Gottschalk, M. Josefiok, J. Jelschen, A. Winter. Removing Energy Code Smells with Reengineering Services. In U. Goltz et al. (eds.) *42. Jahrestagung der Gesellschaft für Informatik e.V.* LNI 208, pp. 441-455, 2012.
- [3] H. Höpfer and C. Bunse. Energy Awareness Needs a Rethinking in Software Development. In *ICSOFT 2011 - Proceedings of the 6th International Conference on Software and Data Technologies*, Seville, Spain, 2011.
- [4] J. Jelschen, M. Gottschalk, M. Josefiok, C. Pitu, and A. Winter. Towards Applying Reengineering Services to Energy-Efficient Applications. In R. Ferenc, T. Mens, and A. Cleve (eds.), *Proceedings of the 16th Conference on Software Maintenance and Reengineering*, 2012.
- [5] M. Josefiok. *An Energy Abstraction Layer for mobile computing Devices*. Masterthesis, University Oldenburg, 2012.
- [6] W. Nebel, M. Hoyer, K. Schröder, D. Schlitt. Untersuchung des Potentials von rechenzentrenübergreifendem Lastmanagement zur Reduzierung des Energieverbrauchs in der IKT, 2009.
- [7] A. Pathak, Y. C. Hu, and M. Zhang. Where is the Energy spent inside my App?: Fine grained Energy accounting on Smartphones with Eprof. In *EuroSys '12: Proceedings of the 7th ACM european conference on Computer Systems*, pp 1-14. ACM, April 2012.
- [8] K. Roy, M. C. Johnson. Software Design for low Power. In W. Nebel, J. P. Mermet (eds.), *Low power design in deep submicron electronics*, pp 433-460. Springer, Berlin, 1997.
- [9] M. Schröder. Erfassung des Energieverbrauchs von Android Apps. Diplomathesis, University Oldenburg, 2013.
- [10] L. Stobbe. Stromverbrauch von Informations- und Kommunikationstechnik in Deutschland. Technical Report BMWi, November 2008.
- [11] L. Stobbe, N. F. Nissen, M. Proske, A. Middendorf, B. Schломann, M. Friedewald, P. Georgieff, T. Leimbach. Abschätzung des Energiebedarfs der weiteren Entwicklung der Informationsgesellschaft Abschlussbericht an das Bundesministerium für Wirtschaft und Technologie. Technical report, Fraunhofer-Institut für System- und Innovationsforschung, 2009.
- [12] L. Zhang, B. Tiwana, R. P. Dick, Z. Qian, Z. M. Mao, Z. Wang, and L. Yang. Accurate online Power estimation and automatic Battery Behavior based Power Model generation for Smartphones. In *IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*, pp 105-114. ACM, 2010.