Measurement 94 (2016) 103–118

Contents lists available at ScienceDirect

Measurement

journal homepage: www.elsevier.com/locate/measurement

An environment for automated measurement of energy consumed by mobile and embedded computing devices

Armen Dzhagaryan [a], Aleksandar Milenković [a,*], Mladen Milosevic [b], Emil Jovanov [a]

[a] Department of Electrical and Computer Engineering, The University of Alabama in Huntsville, 301 Sparkman Drive, Huntsville, AL 35899, USA
[b] Philips Research North America, 2 Canal Park, 3rd Floor, Cambridge, MA 02141, USA

ARTICLE INFO

Article history:
Received 7 November 2014
Received in revised form 22 July 2016
Accepted 25 July 2016
Available online 27 July 2016

Keywords:
Mobile computing
Computerized instrumentation
Current measurement
Energy measurement
Performance evaluation

ABSTRACT

Mobile and embedded computing devices have become the dominant type of computing platforms. Energy-efficiency is a key requirement for these devices, underscored by growing reliance of consumers on services delivered through them and their growing complexity and sophistication. A detailed measurement-based characterization of energy consumed by applications running on mobile and embedded computing devices is important for both device manufacturers and application developers, as it may identify energy-demanding components and activities and guide optimizations. In this paper, we describe an environment for automated energy measurements of applications running on Android mobile and bare embedded computing devices. We discuss hardware and software aspects of the environment and several approaches to runtime capturing and timestamping of activities of interest. Finally, we demonstrate the use of the environment in several case studies conducted on two smartphones and a wearable device.

1. Introduction

Battery-powered computing devices such as mobile devices (smartphones, tablets, e-readers) and wearable consumer devices (e.g., fitness trackers, smart watches, headsets, glasses, smart clothing, wearable cameras) have become the dominant computing platforms for generating and consuming digital information. The number of smartphones and tablets shipped in 2015 reached 1.43 billion and 241 million, respectively, whereas the number of personal computers was ~290 million [1,2]. Modern smartphones and tablets have evolved into powerful computing platforms with significant processing power, storage capacity, myriad of communication interfaces, and numerous sensors. New applications have emerged in areas of communication, navigation, social networking, mobile health, and entertainment. The number of wearable devices shipped in 2015 was 232 million, including 50.4 million smartwatches, 128.5 million Bluetooth headsets, and ~35 million wristbands. Modern wearable devices are rapidly proliferating, offering new applications across the health, fitness, sports, and communication segments. They have a potential not only to empower consumers but also to transform modern business processes and improve worker productivity.

Both mobile and wearable devices employ batteries as the source of energy and share common requirements to be small and lightweight and to provide a rich set of functions. However, battery capacity is directly proportional to its size and weight. Growing dependency of users on services delivered through their battery-powered devices makes their energy-efficient operation a top priority. Energy efficiency is a prime design requirement for device manufacturers and application developers alike. It is driven by several key factors, including (i) limited energy capacity of batteries, (ii) cost considerations favoring less expensive packaging, and (iii) user convenience favoring lightweight designs with small form factors that operate for long periods without battery recharges.

A number of recent research studies have focused on power profiling and power estimation of mobile computing platforms. Carroll and Heiser quantified energy consumption of each component in a mobile device by performing rigorous tests and then simulating a number of usage scenarios on mobile devices [3]. Rice and Hay profiled the energy consumption of connecting and transmitting data over a wireless network [4,5]. Bircher and John used processor performance counters and system-specific models to estimate consumption of CPU, memory, disk, and I/O [6]. Pathak et al. [7,8] and Li and John [9] used system call tracing and known observations of the system to generate models that can perform run-time power estimation with fine-grained measurements.

* Corresponding author.
E-mail addresses: aad0002@uah.edu (A. Dzhagaryan), milenka@uah.edu (A. Milenković), mladem@gmail.com (M. Milosevic), emil.jovanov@uah.edu (E. Jovanov).

http://dx.doi.org/10.1016/j.measurement.2016.07.073
0263-2241/

Runtime power measurements on real mobile devices running common software platforms such as Android, iOS, Tizen, or Windows Phone are important for both researchers and mobile application developers. Measurement frameworks can capture complex interactions between hardware and software stacks that become more and more sophisticated with introduction of systems-on-a-chip (SoCs) with multiple processor cores and a number of customized hardware accelerators. Measurements on real devices can help research studies that target power optimizations or those that target developing analytical models for energy estimation based on parameters derived from real platforms. For mobile developers, adding a power perspective to application debugging and testing may guide optimizations that will result in more energy-efficient mobile applications. Runtime power measurements can be even more critical for wearable devices because they may offer developers deeper insights into energy requirements of both hardware and software components. Thus, system developers may select more energy efficient hardware components and application developers may better exploit energy-efficient operating modes and perform trade-offs between computation and communication tasks.

Whereas several prior studies focused on capturing power traces on smartphones [3–5,10] and wireless sensor network platforms [11], they relied on manual control and post-processing for synchronization of power traces with events in profiled programs or focused on early smartphones and software platforms. In addition, they relied on hardware setups that required inserting a shunt resistor on the power supply line, thus introducing a slight deviation in the power supply of the device under test.

In this paper, we introduce an environment for automated power and energy measurements of modern mobile and wearable devices. Our hardware setup includes a device under test, a National Instruments' (NI) chassis with a battery simulator and a data acquisition device (DAQ), and a workstation. Our custom program running on the workstation interfaces both the device under test and the NI chassis. The program provides (i) a number of configuration options to customize the energy profiling, (ii) functions to remotely control applications and activities executed on the device under test, (iii) functions to synchronize running applications with collecting current samples from the battery simulator, and (iv) scripts for calculating the energy consumed. We describe several approaches to capturing timestamps on devices under test that delimit the profiled activities. The first approach for mobile devices relies on the native Android logging system and does not require any changes in applications that are being profiled. The second approach also relies on the native Android logging system and custom messages inserted in the source code by developers. The third approach relies on CyanogenMod Android and common Linux-like utilities to support launching and timestamping of applications. In addition, we describe an approach for capturing hardware timestamps on bare-metal embedded platforms (platforms that do not have operating system) to synchronize collection of current samples with profiled applications.

Some of the key advantages of the proposed measuring setup are as follows.

- No hardware modifications to mobile devices. The setup requires no hardware modifications or instrumentation of mobile and wearable devices; the device's battery is simply replaced with probes coming from the battery simulator.
- Automated test execution. The measurements are fully automated and controlled by scripts prepared in advance and thus do not require interactive user participation. The scripts can control energy profiling of a number of applications profiled in a single test run.
- Automated synchronization. The workstation and the device under test are time-synchronized using standard network synchronization protocols, thus allowing for precise timestamping of activities of interest. The workstation and bare-metal embedded platforms could optionally be synchronized though the DAQ and lightweight instrumentation of profiled applications.
- High resolution and accuracy. The setup allows collection of up to 200,000 samples per second of power supply current with an accuracy of 1 μA, thus providing a deep insight into inner operations of internal components.

The rest of this paper is organized as follows. Section 2 describes the hardware and software aspects of the setup for energy measurement. Section 3 describes approaches to profiling Android applications, including different methods for collecting timestamps that delimit the activities of interest in time. Section 4 describes an approach to profiling applications running on bare-metal embedded platforms. Section 5 demonstrates the use of the setup in estimating energy-efficiency of several important activities. Section 6 surveys related work, and Section 7 concludes the paper.

2. Measuring setup

Our setup for energy profiling, shown in Fig. 1, consists of a *device under test*, an NI PXIe-4154 *battery simulator* [12], an NI PXIe-6361 *data acquisition (DAQ)* with an external *Shielded I/O Connection Block* (SCB-68), and a *workstation*. Fig. 2 shows a block diagram of the setup with main components and communications between them. The block diagram illustrates two types of devices under test: an Android mobile and a bare-metal embedded device. As examples of mobile devices we use a Google's Nexus 4 smartphone [13] running Android 4.3.2 operating system [14] and an OnePlus One smartphone [15] running CyanogenMod 12.1 [16]. As an example of an embedded device we use a custom wearable platform called *Smart Button* [17] that is developed for automated assessment of mobility in elderly and Parkinson's patients. Whereas the paper focuses on energy profiling of Android platforms and a bare-metal embedded platforms, our hardware setup can be used to profile applications running on other mobile software platforms such as iOS, Tizen, or Windows Phone, as well as on other types of embedded platforms.

The battery simulator and the data acquisition device are connected to an MXI-Express Interface card inside the workstation. The battery simulator is used (i) to power a device under test through probes, thus bypassing the actual battery of the profiled device, and (ii) to measure the current drawn by the device while running applications. The DAQ is used to assert digital outputs or monitor digital inputs that can be connected to the device under test and used to trigger or observe events of interest for power profiling. The workstation runs our custom program called *mLViewPowerProfile* that interfaces (i) the device under test to manage activities and applications that are being profiled, (ii) the battery simulator to capture and record the current sample measurements, and (iii) the DAQ to capture hardware events triggered by software running on embedded platforms. The following subsections shed more light on each component in our setup.

2.1. Devices under test

The Google's Nexus 4 smartphone [17] is powered by a Qualcomm's Snapdragon S4 Pro (APQ8064) SoC [18] that includes a quad-core ARM processor running at up to 1.512 GHz clock and an Adreno 320 graphics processor [19]. Nexus 4 has 2 GB of RAM memory and 16 GB of built-in internal storage. It uses a 4.7 inch