An Energy-Saving Framework for Mobile Devices Based on Crowdsourcing Intelligences

Guangtai Liang, Shaochun Li
IBM Research - China
Beijing, China, 100193
{gtliang, lishaoc}@cn.ibm.com

Abstract

Rich mobile apps make mobile devices increasingly pervasive in our daily life. However, energy consumptions of mobile devices brings lots of users' frustrations. To guarantee good experiences of mobile users, we propose an energy-saving framework for mobile devices, which uses a set of coarse-grained and general-purpose energy-waste heuristics as a starting point and then smartly takes advantages of crowdsourcing intelligence to refine energy-waste related knowledge to help detect/resolve energy wastes in mobile devices. In return, summarized energy-waste related knowledge can be presented to the developers of related mobile apps and guide them to identify/fix related energy bugs. Through initial evaluations, we demonstrate the proposed framework is able to extend the lifetime of a mobile device with one full charge to a large degree (e.g., 30%-70%).

Categories and Subject Descriptors D.2.4 **[Software/Program Verification]**: Reliability, Validation

Keywords: energy-saving framework, mobile devices, crowdsourcing intelligence

1. Introduction

The rapid energy consumption of mobile devices has become an important problem because of the limited energy capacity of mobile devices and the rising complexity of applications running on these devices. Energy problems have led to significant frustrations of mobile users (e.g., 70% of phones returned to Motorola were related to energy problems [1]). Energy problems could happen due to lots of root causes such as programming errors, inappropriate API usage, or external condition changes [2].

To alleviate this problem, many energy-saving solutions have been proposed. Qian and et al. [3] apply energy-aware profiling techniques to identify energy bugs of mobile apps, which requires specific input scenarios to execute apps on mobile devices. Research solutions [4, 5] focus on evaluating energy efficiency of code implementations by monitoring power consumptions of test-suite executions. Pathak and et al. [6] applies static code analyzers to detect particular types of energy bugs (e.g., WakeLock related bugs).

These existing solutions provide "limited" detection or diagnosis capabilities to identify "particular types" of energy bugs in "development" time. However, a lot of energy problems still probably happen in front of huge mobile users. To address this problem, we propose a general-purpose energy-saving framework for mobile devices based on crowdsourcing intelligence. On the one hand, the framework smartly integrates crowdsourcing intelligence (i.e., feedbacks from large user bases) to detect and resolve energy problems happened in devices of mobile users. On the other hand, the framework supports to extract out meaningful energy-waste related knowledge (e.g., patterns) from collected crowdsourcing intelligence, and then provides the knowledge to related development teams of mobile apps to accelerate their fixing processes on energy problems.

2. Framework

Figure 1 shows the overview of the proposed energy-saving framework. In the initial phase, we apply heuristic-based algorithms to detect suspicious energy-waste instances for users' confirmations. Then, based on users' feedbacks (i.e., crowdsourcing intelligence), we continuously improve the effectiveness of the detection and resolution of energy-wastes in mobile devices.

The framework includes two main parts: the user-side mobile app and the cloud-side backend. The mobile app provides "interaction UIs" (responsible for interacting with mobile users), "native environment monitor" (collecting metrics via monitoring native environment), "energy-waste detector" (detecting energy-wastes in mobile devices based on monitoring data), and also "energy-waste resolution engine". The backend includes "energy-waste instances & resolutions" (capturing user-confirmed energy-waste instances and also the resolution actions of users on these instances), "energy-waste pattern recognition" (extracting meaningful patterns for effective energy-waste recognition), and "energy-waste patterns".

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

MobileDeLi'15, October 26, 2015, Pittsburgh, PA, USA
© 2015 ACM. 978-1-4503-3906-3/15/10...$15.00
http://dx.doi.org/10.1145/2846661.2846665

2.1 Energy-Waste Identification and Resolution

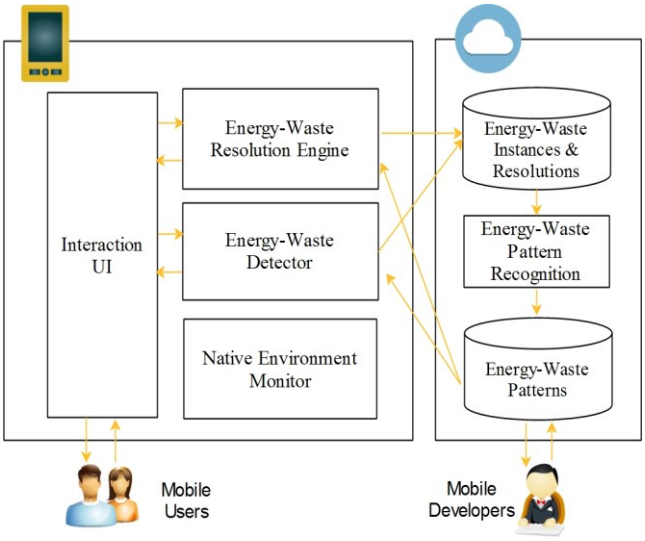

Figure 1. The framework overview.

By monitoring native environments (e.g., running periods, running mode, energy consumptions, version, and acquired sensors of each running app) of users' mobile devices, we first apply coarse-grained heuristics to automatically identify suspicious energy-waste instances (e.g., if a mobile app is transferred into the background but its acquired sensors are not released and its energy consumption does not turn less, we treat the mobile app as a suspicious energy-waste instance). Then we prompt the suspicious energy-waste warnings and potential corresponding resolutions (e.g., to terminate/restart certain apps to avoid further energy drains) to users and collect their confirmations. Since such heuristic-based algorithms may introduce false positives (e.g., for navigation apps, even they are running in the background, they also need to capture the GPS sensor to get users' real-time location to continue the navigation.), in this step, we support another innovative warning-notification way (Selected Notification) to save users' efforts: first send warnings from different users to the central backend; then, the backend groups same warnings into single collections; once the size of a warning collection is larger than a threshold (e.g., 100), the backend randomly selects out 10% users for warning notification; once someone among the 10% users confirms the warning, the backend notify warnings to the left 90% users.

2.1 Intelligence Collection and Analysis

Under users' permissions, we upload and save user-confirmed energy-waste instances and also related resolution actions to the cloud-side dataset "Energy-Waste In-stances & Resolutions". Then, as the "crowdsourcing intelligence" turns enough richer, we would apply pattern recognition techniques to identify the frequent patterns of energy-waste instances and resolutions (e.g., what specific apps under what usage scenarios will be treated as energy waste instances, and also what corresponding actions users would make to remediate these energy instances). Such frequent patterns provide high-value (i.e., user-confirmed, frequent) knowledge for effectively identifying real-world energy-waste instances and also recommending remediation solutions of users' interests.

2.2 Intelligence Collection and Analysis

In this framework, "energy-waste patterns" represent meaningful and valuable knowledge (i.e., crowdsourcing intelligence) on real-world energy problems that happen in front of real-world mobile users. In this framework, we apply the knowledge along two different ways to accelerate the resolution of related energy problems: 1) User-Side Problem-Remediation Optimization. We continuously apply extracted knowledge to improve the effectiveness of the client-side identification and resolution of energy wastes. With more and more crowdsourcing intelligence generated, the framework can identify real-world energy wastes of users' interests more and more effectively. 2) Developer-Side Bug-Fixing Acceleration. We continuously push energy-waste patterns to related development teams of mobile applications. Based on the guidance of user-confirmed energy-waste instances, developers can identify related code-level energy bugs with less efforts.

3. Conclusion

In this paper, we present a general-purpose energy-saving framework for mobile device based on crowdsourcing intelligence. We implemented a prototype (an android app and a Bluemix-based backend service) and made initial evaluations. The evaluations show that our framework can effectively extend the energy life of mobile devices around 30-70%. However, there are still lots of future works for us to further improve the effectiveness of the framework (e.g., applying big data techniques to mine energy-waste patterns more effectively and efficiently, providing proper interaction mechanisms for developers to assistant them in detecting and fixing related energy bugs).

References

[1] "The bulk of returned motorola phones are the result of applications that cause performance/energy problems." URL: http://tinyurl.com/44hns9p

[2] A. Pathak, Y. C. Hu, and M. Zhang. Bootstrapping energy debugging on smartphones: a first look at energy bugs in mobile devices. HotNets 2011.

[3] F. Qian, Z. Wang, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck. Profiling resource usage for mobile applications: a cross-layer approach. MobiSys, 2011.

[4] W. Baek and T. M. Chilimbi. Green: a framework for supporting energy-conscious programming using controlled approximation. PLDI, 2010.

[5] I. Manotas, L. Pollock, and J. Clause. SEEDS: A software engineer's energy-optimization decision support framework. ICSE, 2014.

[6] A. Pathak, A. Jindal, Y. C. Hu, and S. P. Midkiff. What is keeping my phone awake?: characterizing and detecting no-sleep energy bugs in smartphone apps. MobiSys 2012.