Pervasive and Mobile Computing 20 (2015) 47–63

Contents lists available at ScienceDirect

Pervasive and Mobile Computing

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

A middleware framework for application-aware and user-specific energy optimization in smart mobile devices

Sudeep Pasricha [a,*], Brad K. Donohoo [b], Chris Ohlsen [c]

[a] Colorado State University, Fort Collins, CO 80523, USA

[b] U.S. Department of the Air Force, Roy, UT 84067, UT 84067, USA

[c] Woodward, Inc., Fort Collins, CO 80525, USA

ARTICLE INFO

Article history:
Received 5 August 2014
Received in revised form 22 December 2014
Accepted 7 January 2015
Available online 14 January 2015

Keywords:
Energy optimization
Smart mobile systems
Pervasive computing
Machine learning
Middleware

ABSTRACT

Mobile battery-operated devices are becoming an essential instrument for business, communication, and social interaction. In addition to the demand for an acceptable level of performance and a comprehensive set of features, users often desire extended battery lifetime. In fact, limited battery lifetime is one of the biggest obstacles facing the current utility and future growth of increasingly sophisticated "smart" mobile devices. This paper proposes a novel application-aware and user-interaction aware energy optimization middleware framework (*AURA*) for pervasive mobile devices. *AURA* optimizes CPU and screen backlight energy consumption while maintaining a minimum acceptable level of performance. The proposed framework employs a novel Bayesian application classifier and management strategies based on Markov Decision Processes and Q-Learning to achieve energy savings. Real-world user evaluation studies on Google Android based HTC Dream and Google Nexus One smartphones running the *AURA* framework demonstrate promising results, with up to 29% energy savings compared to the baseline device manager, and up to 5×savings over prior work on CPU and backlight energy co-optimization.

1. Introduction

Mobile smartphones and other portable battery operated systems (PDAs, tablets) are pervasive computing devices that have emerged in recent years as essential instruments for communication, business, and social interactions. At the end of 2013, there were more than 6.8 billion mobile subscribers worldwide, with smartphone sales showing the strongest growth. Over 1,600,000 apps have been developed across various mobile platforms, with the Apple App Store and Google Play being the largest app stores. Popular mobile activities include web browsing, multimedia, games, e-mail, and social networking [1]. Overall, these trends suggest that mobile devices are now the new development and computing platform for the 21st century.

Performance, capabilities, and design are all primary considerations when purchasing a smart mobile device; however, battery lifetime is also a highly desirable attribute. Most portable devices today make use of lithium-ion polymer batteries, which have been used in electronics since the mid 1990s [2]. Although lithium-ion battery technology and capacity has improved over the years, it still cannot keep pace with the power consumption demands of today's mobile devices. Until a new battery technology is discovered, this key limiter has led to a strong research emphasis on battery lifetime extension, primarily using software optimizations [3–14].

* Corresponding author.
E-mail addresses: sudeep@colostate.edu (S. Pasricha), brad.donohoo@gmail.com (B.K. Donohoo), chris.ohlsen@gmail.com (C. Ohlsen).

http://dx.doi.org/10.1016/j.pmcj.2015.01.004
1574-1192/

It is important to note that outside of the obvious differences between mobile devices and a general PC – weight and size, form factor, computational capabilities, and robustness – a key difference can be found in the user interaction patterns and interfaces. Unlike a desktop or notebook PC in which a user typically interacts with applications using a pointer device or keyboard, applications on mobile devices most often receive user input through a touch screen or keypad events. Many times applications are interacted with for short durations throughout the day (e.g., few seconds or minutes instead of hours) and these patterns are often unique to each individual user. Significant differences in user interaction patterns make a general-purpose power management strategy unsuitable for mobile devices.

In this work, we present a novel application and user interaction aware energy management framework (*AURA*) for pervasive mobile devices, which takes advantage of the user-specific patterns of interactions with applications running on mobile devices to optimize CPU and backlight energy consumption. In order to balance energy consumption and quality of service (QoS) requirements that are unique to each individual user, *AURA* makes use of a Bayesian application classifier to dynamically classify applications based on user interaction. Once an application is classified, *AURA* utilizes Markov Decision Process (MDP) or Q-Learning based power management algorithms to adjust processor frequency and screen backlight levels to reduce system energy consumption between user interaction events. Overall, we make the following novel contributions:

- We conduct usage studies with real users and develop a Bayesian application classifier tool to categorize mobile applications based on user interaction activity;
- We develop an integrated MDP/Q-Learning based application and user interaction-aware energy management framework that adapts CPU and backlight levels in a mobile device to balance energy consumption and user QoS;
- We characterize backlight and CPU power dissipation on Android OS based HTC Dream and Google Nexus One smartphone architectures;
- We implement our framework as middleware running on the HTC Dream and Google Nexus One smartphones and demonstrate real energy savings on commercial apps running on the devices.
- We perform real-world user evaluation studies with the Google Android based HTC Dream and Google Nexus One mobile devices running the *AURA* framework and demonstrate promising results, with up to 29% energy savings compared to the baseline device manager; and up to $5\times$ savings over the best known prior work on CPU and backlight energy co-optimization, with negligible impact on user quality of service.

This work is a significantly extended version of our previously published conference paper [15] with the following major additions: (i) additional power models, algorithm analysis, and results for a new phone architecture—the Google Nexus One; (ii) analysis and results for a new power management algorithm based on Q-Learning; (iii) a finer-grained app classification that allows the algorithms to be tuned to each application more precisely; (iv) a more extensive user–device interaction field study in which we examine the variability of user interaction patterns in Section 2; (v) further details in Section 3.2 about the activation of DVFS during idle periods; (vi) a study of the effect of successful prediction rates on actual user satisfaction and definition of a minimum acceptable performance level, in Section 5; (vii) a study on the performance of the power management algorithms when more mispredictions are allowed, in Section 5; and (viii) more comprehensive related work in Section 6 explaining how our work is different and novel in comparison with previously published work.

2. AURA energy management framework

In this section, we present details of the *AURA* framework. In Section 2.1 we first describe our fundamental observations that lay the foundation for energy savings in mobile devices. Section 2.2 presents results of field studies involving users interacting with apps on mobile devices. Section 2.3 gives a high level overview of the *AURA* middleware framework. Subsequent sections elaborate on the major components of the framework.

2.1. Fundamental user–device interaction mechanisms

Here we explain the underlying concepts stemming from the psychology of user–device interactions that drive the CPU and backlight energy optimizations in the *AURA* framework.

There are three basic processes involved in a user's response to any interactive system [16], such as a smartphone or a personal computer. During the *perceptual* process, the user senses input from the physical world. During the *cognitive* process, the user decides on a course of action based on the input. Finally, during the *motor* process, the user executes the action with mechanical movements. These three processes are consecutive in time and can be characterized by an *interaction slack*. For instance, when interacting with an app on a mobile device, the time between when a user interacts with an app (e.g., touching a button) and when a response to that interaction is perceptible to the user (e.g., the button changes color) is the perceptual slack; the period after the system response during which the user comprehends the response represents the cognitive slack; and finally the manual process of the next interaction (e.g., moving finger to touch the screen) involves a motor slack. Fig. 1 illustrates this process. Before the user physically touches the input peripherals, the system is idle during the cumulative slack period, for hundreds to possibly many thousands of milliseconds. During this time, if the CPU frequency can be reduced, energy may be saved without affecting user QoS. **CPU energy** optimizations in *AURA* exploit this inherent slack that arises whenever a user interacts with a mobile device.