Analyzing Power Consumption and Characterizing User Activities on Smartwatches : Summary

Abstract—In this paper, we study real end-user data by releasing a logger application that monitors users while they are using their watches. We collect traces of real smartwatch user activities from 32 users (4 brands, 7 different models) with totally 70 days of continuous use. We use the traces to characterize usage and power consumption while identifying operational bottlenecks. Specifically, we develop regression-based power models using system-level hardware components and analyze high-level characteristics: general usage behavior, interaction with the battery, total and component-wise power consumption, network activity, and power consumption of frequently run applications. Our models provide insights about the power breakdown among the hardware components. We present 9 major findings and observations including: (1) battery management is important: majority of users charge their devices every 15 hours; (2) screen and CPU consume more than half of the active power: on average, they use 38.5% and 32.6% of the active power, respectively; and (3) application characteristics affect power consumption: downloaded applications (from third party developers) consume up to 4 times more power than in-built default applications.

Keywords— Smartwatch; component-wise power consumption; system level power modeling; system analysis; usage behavior; network characteristic

I. INTRODUCTION

Smartwatch industry is growing rapidly [1,2], but little is known about the usage and power consumption characteristics of these battery-driven devices. In this paper, we show tools and methods for studying the power consumption and usage characteristics of smartwatches. Because despite the importance of the end user activity in understanding workloads, to the best of our knowledge, there does not exist any study for analyzing energy consumption or characterizing user behavior at a system-level on smartwatches. We develop a logger application for Android Wear smartwatches that logs user activities and sends traces back to our servers through their connected phones. We release the logger into the market to collect logs of real anonymized users on real smartwatch devices in real environments. We then demonstrate how these logs can be used to analyze power consumption and characterize usage behavior. Specifically, when analyzing the power consumption, we develop linear regression-based power estimation models for each user, which use system-level hardware components. We chose to use linear regression, because it provides insights about power breakdown by bringing out each component's contribution to total power consumption through their coefficient values.

Overall, we present several findings and observations along with their implications on energy-efficient design and optimization by analyzing of real smartwatch usage of 32 users with their own smartwatches. Our main contributions and findings can be listed as follows:

Power Modeling

- We develop linear-regression-based power estimation models to inspect component-wise power consumption. In these models, we use easily accessible measurements to accurately predict the system-wide power consumption of various smartwatch architectures.

Battery Charging

- Users charge their batteries frequently (typically more than once a day) and rarely use it more than 15 straight hours. Also they generally use their watches until the remaining battery level is low (less than 30%).

Component-wise Power Consumption

- There is significant variation in the usage behavior. This results in a high variation on power consumption across users.
- Active state consumes 53.2% of the total system power on average across all brands; even though it only accounts for 11% of the usage time. Of the hardware components, the screen and the CPU consume the most power across all brands; 38.5% and 32.6% of the active power consumption on average, respectively.
- Across all users, the watch is in the Idle state 89% of the time and this accounts for 46.8% of the total system power.
- Most users do not switch between multiple screen brightness levels, nor do they use automatic brightness adjustment.
- The watch's CPU is either above 70% or below 30% utilization level 64.1% of the *Active* time.

Networking

- Bluetooth and WiFi (for the watches with Android Lollipop) are the two possible technologies to connect to phone or to environment. Most users do not change the default inactive WiFi state and do not use it. Bluetooth is the main and most used communication method.

Applications

- CPU utilization of some applications developed by third parties is significantly higher than others. Of the broad perspective applications that use screen in the name of "watchface applications" use CPU more than others.

General Implications

We use the Weka-tool [3] to create our linear models to predict total and component-wise power consumption in various smartwatches. We also demonstrate that power consumption can be accurately predicted in system level. Across all users, we see battery management is a significant part of the smartwatch user experience, since our target devices need daily charges. It is critical to perform real user studies when studying smartwatches because we see large variation in our users' power consumptions. We demonstrate that when the watch is in active usage, it consumes the majority of the power, even though it accounts for a small fraction of the total time. The screen and CPU require the most attention with respect to energy efficiency. Guiding users more to use automatic brightness options in the watch could be used for saving power. BLE (Bluetooth low energy) technology used in all our target devices and it keeps the network energy consumption low [4] when compared to other hardware components. We see no WiFi usage in our logs. Finally, we observe some applications downloaded from Android Market consume more power than default applications. Users can be warned about the average power consumption of the applications before they install.

II. METHODOLOGY

Android Wear Smartwatches

Our target smartwatches in this paper are all smartwatches that support the open source Google Android Wear platform [7]. At the time of our studies, there were 5 different brands and 10 different models that were using Android Wear OS. Among them, we study 4 different brands and 7 different models ($Table$ 1). The software on the smartwatches are the open-source Android Wear platform, which consists of a slightly modified 5.1 Lollipop Linux kernel, and a general framework of C, C++, and Java code. The framework includes The Android Runtime Environment (ART) that replaces Dalvik VM within Android 5.0, a variant of Java implemented by Google. All user space applications are ART executable, which run in an instance of the ART environment. [5]

Logging User Activity

To study the usage of the Android Wear Smartwatches, we have developed a logger application that logs user activity events, as well as system-level performance measurements. The logger is developed as a normal ART executable using the Java standard libraries available in the Android and Android Wear framework. Thus, it runs on all Android Wear Smartwatch devices without any special hardware or OS support. At a high-level the logger application consists of two parts: (1) a GUI application which looks like a normal Android Wear GUI application and (2) an associated background service to provide logging functionality. We release the logger on Android Market, the portal for Android Wear users to browse and download Android Wear applications through their connected phones.

Building the Linear Model

We use real-time measurements of our target smartwatches to develop linear regression-based power estimation models. The main idea is to find the total power consumption as well as its distribution on collected system-level hardware components. Therefore, while the inputs to the models are the statistics collected from the users' watches ($Table$ 1), outputs are the total power consumption. The raw data samples are collected as average values at 1 Hz time interval to reduce perturbation on the system and minimize the execution and power consumption of the logger application. The collected statistics and the measured power consumption are fed into the Weka-tool (Weka 3-6-1 version) to find the regression coefficient c_j for each parameter. We use a 10-fold cross-validation while building the models; hence the training data and the test data do not have any overlap. Using these coefficients and the logged data, we can predict each device's power consumption by simply providing the statistics for them. Also, within these coefficient values, each individual hardware component's power consumption can be estimated in linear model.

TABLE I. STUDIED SMARTWATCH BRANDS AND MODELS ALONG WITH THEIR COLLECTED HARDWARE COMPONENTS FOR POWER MODELS.

Brand/Model	***Collected Statistics***
Asus Zen Watch 2, Huawei Watch, LG Watch R, LG Watch Urbane, Moto 360 2nd Generation	Bluetooth on/off, Bluetooth data transfer, Screen on/off, Screen brightness, CPU utilization, SD card read/write, WiFi on/off, WiFi data transfer
LG Watch G	Bluetooth on/off, Bluetooth data transfer, Screen on/off, Screen brightness, CPU utilization, SD card read/write
Moto 360	Bluetooth on/off, Bluetooth data transfer, Screen on/off, Screen brightness, CPU utilization, CPU frequency, SD card read/write, Interrupts (5 interrupts), WiFi on/off, WiFi data transfer

REFERENCES

[1] Allied Market Research Report: Global Smartwatch Market - size, Industry Analysis, Trends, Growth and Forecast, Jan 2015

[2] EMarketer Report: Putting Smart Watch Adoption in Perspective, June 11, 2015

[3] http://www.cs.waikato.ac.nz/ml/weka

[4] C.Gomez, J.Oller, J.Paradells. Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology, Sensors, 2012

[5] https://www.android.com/wear/