The Missing Numerator: Toward a Value Measure for Smartphone Apps

Anudipa Maiti and Geoffrey Challen
Department of Computer Science and Engineering
University at Buffalo
{anudipam,challen}@buffalo.edu

ABSTRACT

While great strides have been made in measuring energy consumption, these measures alone are not sufficient to enable effective energy management on battery-constrained mobile devices. What is urgently needed is a way to put energy consumption into context by measuring the *value* delivered by mobile apps. While difficult to compute, an accurate value measure would enable cross-app comparison, app improvement, energy inefficient app detection, and effective runtime energy allocation and prioritization. Our paper motivates the problem, describes requirements for a value measure, discusses and evaluates several possible inputs to such a measure, and presents results from a preliminary (unsuccessful) attempt to formulate one.

1. INTRODUCTION

Measuring app energy consumption[1] on mobile devices is nearly a solved problem. This is due to great strides made in both generating and validating energy models that deliver accurate runtime energy consumption estimates [4, 11, 8, 7, 12] and in accurately attributing energy consumption, even for asynchronous and shared resources [10, 2]. Accurate energy models bring us closer to the goal of effective energy management on battery-constrained devices.

But accurate energy measurement alone is not enough, because even perfectly-accurate measurements of energy consumption are insufficient to answer critical energy-related questions faced by users and developers, including:

- Which of the following two apps is more energy efficient?
- Will this change to an app make it more energy efficient?
- Is a particular app an *energy virus*?
- How should the limited energy resources on a given app be prioritized?

[1]To avoid confusion between app and energy usage, we use *consumption* exclusively when referring to energy usage and *usage* exclusively when referring to user interaction with apps.

HotMobile'15, February 12–13, 2015, Santa Fe, NM, USA
ACM 978-1-4503-3391-7/15/02. . .$15.00.
http://dx.doi.org/10.1145/2699343.2699360

Unifying all of these questions is one missing component: a measure of app *value*, which can be used alone or combined with energy consumption to compute energy *efficiency*:

$$\frac{value}{energy}$$

Armed with a measure of value we can return to the difficult questions posed above. By computing efficiency users can perform apples-to-apples comparisons of apps in order to evaluate two video conferencing tools, web browsers, or email clients. Developers can determine whether a new feature delivers value more or less efficiently than the rest of their app and better understand the differences in energy consumption across different users. Measuring value allows a rigorous definition of an *energy virus* as an app that delivers little or no value per joule, and for systems to reward efficient apps by prioritizing limited resources based on app value or energy efficiency. After all the progress we have made in computing the denominator—energy consumption—we believe that the search for the missing numerator is the most important open challenge in energy management.

Developing such a measure, however, is difficult. To be effective it must work across almost the entire spectrum of smartphone apps, which represent an incredible diversity of different goals, interfaces, and interaction patterns. It must also work across a variety of different users with different usage patterns. It must be efficient to compute, since it should not compete for the same limited energy resources that it is intended to help manage. Ideally it should require little to no user input, since this will make it burdensome and error-prone. And to make matters worse, there is no obvious way to measure ground truth to compare against—even in a lab. Despite all these challenges, however, even a semi-accurate value measure would greatly benefit energy management on battery-constrained smartphones. With users continuing to report battery lifetime as their top concern with smartphones [9], we believe this effort is worthwhile.

In this paper we motivate the idea of a value measure and describe an early failure at developing one. We begin in Section 2 by describing how useful such a measure would be while also formulating design requirements for the value measure itself. Section 3 presents an overview of possible inputs into such a measure and discussion of how each could be measured and how useful it might be. In Section 4 we present our initial effort at formulating a value measure based on content delivered through the video display and audio output—an attempt that we consider a failure based on the result of a user survey, but a failure that we hope sheds some light on this difficult challenge.

2. USES AND REQUIREMENTS

To motivate the need for a value measure, we return to the questions posed in the introduction and explore each in more depth. These use cases also help us develop requirements for our measure, which are summarized at the end of this section. We begin by exploring the basic question at the heart of the problem: what is the value of an app?

2.1 What is App Value?

All smartphone users intuitively realize that smartphone apps differ in value—an email client, for example, is probably more valuable than a app that makes random sounds. But is it possible to quantify these subjective distinctions and produce a value measure? To argue that this is possible we present two experiments that elucidate smartphone app value in the form of both ordinal and cardinal utilities:

1. You will be required to remove some number of apps from your smartphone. Order the apps you are currently using from least important to most important. The N least important apps will be removed.
2. You will be required to create an energy budget for the apps you use on your smartphone. During any discharging cycle, once an app runs out of energy you will not be able to use it until you plug in your smartphone. Allocate battery percentages to each app you use.

We plan to engage smartphone users in studies to explore in more detail which of these approaches is more effective, comparing them by comparing users' levels of satisfaction under each scenario. In the first experiment we ask users to uninstall apps because often apps have a background component that keeps consuming energy even when the app is no longer being used. For our value measure we are hopeful that users will prove capable of assigning cardinal utilities to apps—as in the second experiment—since this matches most directly with our proposed value measure and could provide ground truth for a value measure computed automatically. The second experiment also engages users directly in the task of allocating energy, which is one way that a value measure could be used. However, if ordinal utilities prove more intuitive we can still compare the ordering generated by our measure with the ordering generated by users, although the values of the measure will still require justification.

In either case, we believe that these experiments do suggest the existence of quantifiable value for smartphone apps. We are not claiming, however, that these setups are the only way or the right way to measure value. In both cases low value measures have fairly extreme consequences—the app is actually removed or rendered unusable. This may cause users to overvalue essential tools such as communication apps and undervalue inessential apps that nevertheless provide them with a great deal of enjoyment such as games. However, given that our goal is a value measure that can be paired with and used to allocate energy, and that energy exhaustion has such severe consequences on the usability of all apps, a more extreme experimental setup may be justified.

2.2 Comparing Apps

With some confidence that smartphone app value can be quantified, we now proceed to motivate the idea of a value measure by discussing several ways in which it could be used.

The most powerful use of a value measure would be to compare apps by comparing their energy efficiency, therefore overcoming the most critical flaw in current attempts to compare or categorize apps by their energy consumption alone [6]. Consider attempting to compare a chat client and video conferencing app by only measuring their energy consumption. Unless it is terribly written, the chat client will consume less energy. But this does not mean that it is efficient, or that the video conferencing app is not. Ultimately, all the energy consumption comparison truly reveals is that the two apps do different things—which we already knew.

Using energy consumption alone even makes apples-to-apples comparison of the same app difficult. Given an app that consumes twice as much energy on Alice's smartphone than on Bob's, the question of why is left unanswered by pure energy measures. Even if usage time can be used to normalize the comparison, power consumption alone cannot incorporate differences due to the different app features or app configurations used by Alice and Bob.

By computing value and, thus, energy efficiency, we can overcome these weaknesses. A value measure should allow us to compare the efficiency of two apps in different categories based on how efficiently they use energy to deliver user value. Comparisons within the same app category should allow users to select the most efficient email client or web browser. Aggregating results over all users, differences in app energy efficiency should reflect how well the app is written and how well it predicts and adapts to users, not just differences in the core features it provides. When comparing two users using the same app, differences in efficiency should reflect differences in app configurations or app features.

2.3 Evaluating App Changes

A second use for the value measure is helping developers improve their apps and deliver more value per joule. Today's energy profiling tools may be able to show the energy impact of adding a new feature or changing the way that a particular feature is implemented, but energy consumption alone is not sufficient to apply Amdahl's Law properly to the problem of improving app energy efficiency. Developers should strive to make the parts of their app that generate a large amount of value as energy-efficient as possible, remove parts that generate little value while consuming a great deal of energy, and defer work on everything else.

2.4 Detecting Energy Viruses

A measure of app value makes it possible to produce a rigorous definition of the term *energy virus*: an app that produces little to no value per joule. The choice of threshold will require some study, as it is probably impossible to produce a single efficiency cutoff that cleanly separates malicious apps from ones that are merely poorly-written. This definition of energy virus can also be made on a per-user basis. This is important since a non-malicious but poorly-written app that continues to consume energy even long after the user has stopped using it—and it has stopped providing value—functions as an energy virus for that user, but may not for a user that interacts with it more frequently.

2.5 Prioritizing System Resources

An app value measure should be able to be used to prioritize limited system resources, particularly energy but also storage, memory, networking bandwidth and processor time. While mechanisms differ, most previous attempts to control energy consumption rely on some form of rate control which allocates a rate to each app and enforces that rate by slowing or stopping the app when it exceeds its allocation [1, 10, 13, 3]. However, all of these previous efforts have ignored the critical question of how rates should be set. No matter how effective the enforcement mechanisms are, systems that rely on rates will fail if they provide the same rate to Skype and Snapchat, or to a very efficient app and an energy virus.

A measure of value can be used alone or in conjunction with energy consumption to help prioritize limited energy resources. The simplest approach is to attempt to enforce an energy allocation based on the relative value assigned to each app. To encourage apps to be more energy efficient, it may also be beneficial to weight allocations by their energy efficiency, providing a boost to apps that provide a larger amount of value per joule. While there are likely many ways to combine energy consumption with a value measure in order to prioritize energy consumption, it is not clear that energy consumption can be prioritized effectively without some measure of value. The same approach can also be applied to determine how much of any limited system resource to allocate to each app, Together these resource allocation measures can be designed to ensure that high-value apps run smoothly at the expense of lower-value apps.

2.6 Summary of Requirements

The use cases above give rise to a set of requirements for a possible value measurement:

- It should enable aggregate comparisons between apps across categories and users.
- It should enable comparisons between the same app across users or inputs, requiring that it be calculable given data from a single user.
- It should enable targeted development by highlighting what parts of an app generate value and what parts do not.
- It should be efficiently computable without unduly consuming the resources that it is designed to help manage.
- It should be derived with little to no input from the user.

3. VALUE MEASURE INPUTS

To continue we discuss possible inputs to a value measure and how to collect them at runtime. In each case, we also discuss how such statistics could be misleading.

3.1 Overall Usage

There are a variety of different ways to measure overall app usage that could be useful inputs to our value measure. Total foreground time is straightforward to measure, particularly on today's smartphones where one app tends to dominate the display. However, next-generation smartphone platforms that provide multiple apps with simultaneous access to the display will complicate this task by making it more difficult to determine which app the user is paying attention to. Number of starts is also a potentially-useful input, as may be the distribution of interaction times across all times that the app was brought to the foreground.

While these measures of contact time are intuitive, there are obvious cases in which they fail, particularly for apps that spend a great deal of time running in the background in order to deliver a small amount of useful foreground information—such as a pedometer app.

3.2 User Interface Statistics

Patterns of interaction may also be useful to observe, and inputs such as keystrokes and touchscreen events are simple to track. However, there is more obvious differentiation between app interaction patterns between categories—users deliver far more keystrokes to a chat client than to a video player—so interaction statistics will have to be used in conjunction with complementary value measure components that offset the differences between high-interaction and low-interaction apps. This approach also fails in the case where apps deploy confusing or unnecessary interfaces that require a great deal of unnecessary interaction to accomplish simple tasks. Clearly, such apps should not be rewarded.

3.3 Notification Click-Through Rates

Another interesting statistic that could provide insight on app value is how often users view or click through app notifications. When notifications are delivered but not viewed, then it is unclear whether the app needed to deliver them. When clickable notifications—such as those for new email—provide a way for users to immediately launch the app, the percentage of notifications that are clicked versus ignored could be used to at least evaluate how effective the notifications are, and may also reflect on overall app value.

Notification view and click-through rates also help put into context the energy used by apps when they are running in the background. Legitimate background energy consumption should be for one of two purposes: (1) to prepare the app to deliver more value the next time it is foregrounded, as is the case when music players download songs and store them locally to reduce their runtime networking usage; or (2) to deliver realtime notifications to the user. The effectiveness of background energy consumption to fill caches will be reflected in the apps overall energy usage, since retrieving local content is more energy efficient than using the network. Effectiveness of background consumption to deliver notifications may be reflected in the rate at which notifications are viewed or clicked, since a notification that is not consumed did not need to be retrieved.

However, in some cases apps may do an effective job at summarizing the event within the notification itself, providing no need for the user to bring the app to the foreground. Clearly, such apps should not be penalized.

3.4 Content Delivery

Another approach to measuring value that we feel is promising is to consider apps as content delivery agents and measure how efficiently they deliver information to and from the user. Encouragingly, multiple apps that we have previously considered can fit into this framework:

- **Chat client:** the content is the messages exchanged by users, and efficiency is determined by the amount of screen time and interaction required to retrieve and render incoming messages and generate outgoing messages as replies. Value is measured by the content of the messages. Efficient chat clients exchange many messages per joule.
- **Video player:** the content is the video delivered to the user and efficiency is determined by the amount of network bandwidth and processing needed to retrieve and render the video. Value is measured by the information delivered by the videos and efficient video players present a large amount of video content to their users per joule.

- **Pedometer:** the content is the count of the number of steps presented to the user and efficiency is determined by the accelerometer rate and any post-processing required to produce an accurate estimate. Value is measured as the ability to maintain the step count and efficient pedometers can achieve more accuracy in computing values per joule.

However, while this framework is conceptually appealing, fitting each app into it requires app-specific features that we are trying to avoid: content is measured in messages for the chat client, frames for the video player, and the step value accuracy for the pedometer. This raises the question of whether a single measure of content delivery requiring no app-specific knowledge can be utilized in all cases. We explore this question in more detail, as well as differences between the other value measure inputs we have discussed, through the experiment and results described next.

4. RESULTS

To examine the potential components of a value measure further, we utilize a large dataset of energy consumption measurements collected by an IRB-approved experiment run on the PhoneLab testbed. PhoneLab is a public smartphone platform testbed located at the University at Buffalo [5]. 220 students, faculty, and staff carry instrumented Android Nexus 5 smartphones and receive subsidized service in return for willingness to participate in experiments. PhoneLab provides access to a representative group of participants balanced between genders and across a wide variety of age brackets, making our results more representative.

Understanding fine-grained energy consumption dynamics required more information than Android normally exposes to apps. In addition, to explore components of our value measure we also wanted to capture information about app usage—including foreground and background time and use of the display and audio interface—that was not possible to measure on unmodified Android devices. So to collect our dataset we took advantage of PhoneLab's ability to modify the Android platform itself. We instrumented the `SurfaceFlinger` and `AudioFlinger` components in the Android platform to record usage of the screen and audio, and altered the ActivityManagerService package to record energy consumption at each app transition. This allows energy consumption by components such as the screen to be accurately attributed to the foreground app, a feature that Android's internal battery monitoring component (the Fuel Gauge) lacks. Changes were distributed to PhoneLab participants in November 2013 via an over-the-air (OTA) platform update. The resulting 2 month dataset of 67 GB of compressed log files represents 6806 user days during which 1328 apps were started 277,785 times, and used for a total of 15,224 hours of active use by 107 PhoneLab participants.

Our analysis begins by investigating several components of a possible value measure and shows the effect of using each to weight the overall energy consumed by each app. Next, we formulate a simple measure of content delivery by measuring usage of the screen and audio output devices and test it through a survey completed by 47 experiment participants. Unfortunately, our results are inconclusive and open to several possible interpretations which we discuss. We present our results in tabular format where for each measure we rank 10 best performing and 10 worst performing apps in descending order.

4.1 Total Energy

Clearly, ranking apps by total energy consumption computed by adding all foreground and background energy consumption over the entire study says much more about app popularity than it does about anything else. Table 1a shows the top and bottom energy-consuming apps over the entire study. As expected, popular apps such as the Android Browser, Facebook, and the Android Phone component consume the most energy, while the list of low consumers is dominated by apps with few installs. This table does serve, however, to identify the popular apps in use by PhoneLab participants, and as a point of comparison for the remainder of our results.

4.2 Power

Computing each app's power consumption by scaling their total energy usage against the total time they were running, either in the background or foreground, reveals more information, as shown in Table 1b. Our results identify Facebook Messenger, Google+, and the Super-Bright LED Flashlight as apps that rapidly-consume energy, while the Bank of America and Weather Channel apps consume energy slowly. Differences between apps in similar categories may begin to identify apps with problematic energy consumption, such as contrasting the high energy usage of Facebook Messenger with other messaging clients such as WhatsApp, Twitter, and Android Messaging.

4.3 Foreground Energy Efficiency

Isolating the foreground component of execution time provides a better measure of value, since it ignores the time that users spend ignoring apps. Table 1c shows a measure of energy efficiency computed by dividing total foreground energy consumption by total foreground time of an app. Some surprising changes from the power results can be seen. A number of apps have remained in their former categories: Bank of America, which was identified as a low-power app, is also a highly-efficient app when using foreground time as the value measure; and Facebook Messenger, which was identified as a high-power app, is also marked as inefficient. Other apps, however, have switched categories. ESPN Sportscenter and Yahoo Mail do not consume much power, but also don't spend much time in the foreground; interestingly, none of the high-power apps looked better when their foreground usage was considered.

4.4 Content Energy Efficiency

Finally, we use the data we collected by instrumenting the `SurfaceFlinger` and `AudioFlinger` components to compute a simple measure of content delivery. We measure the audio and video frame rates and combine them into a single measure by using bit-rates corresponding to a 30 fps YouTube-encoded video and 128 kbps two-channel audio, with the weights representing the fact that a single frame of video contains much more content than a single sample of audio. We use this combined metric as the value measure and again use it to weight the energy consumption of each app, with the results shown in Table 1d.

Comparing with the foreground energy efficiency again shows several interesting changes. Yahoo Mail, which foreground energy efficiency marked as inefficient, looks more efficient when content delivery is considered. While it is possible that one PhoneLab participant uses it to read email

Rank	App	Energy (As)
1	Android Browser	41052.703
2	Facebook	37268.388
3	Chrome Browser	22719.020
4	Android Phone	18122.433
5	Gmail	17402.896
6	Android Messaging	17342.926
7	WhatsApp Messenger	16467.477
8	Google Search	15370.252
9	Candy Crush Saga	12767.649
10	Android Gallery	11050.363
10	Google+	586.586
9	Android Calculator	449.474
8	NFL Mobile	344.492
7	UB Parking	311.766
6	Super-Bright LED Flashlight	218.870
5	Starbucks	174.609
4	Google Keep	174.263
3	Dropbox	160.939
2	ESPN SportsCenter	108.965
1	Bank of America	98.007

(a) **Most and Least Energy-Consuming Apps.**

Rank	App	Consumption Rate (A)
1	Facebook Messenger	0.774
2	Google+	0.614
3	Super-Bright LED Flashlight	0.600
4	UB Parking	0.598
5	Android Music	0.446
6	Google Search	0.428
7	NFL Mobile	0.386
8	Pandora	0.326
9	Starbucks	0.282
10	Android News and Weather	0.254
10	Chrome Browser	0.099
9	WhatsApp Messenger	0.095
8	Twitter	0.078
7	Yahoo Mail	0.077
6	Android Messaging	0.061
5	Skype	0.040
4	YouTube	0.036
3	ESPN SportsCenter	0.021
2	The Weather Channel	0.019
1	Bank of America	0.011

(b) **Fastest and Slowest Energy-Consuming Apps.**

Rank	App Name	Efficiency
1	Bank of America	83.717
2	The Weather Channel	49.861
3	Skype	23.779
4	YouTube	19.880
5	Android Messaging	12.933
6	Android Gallery	9.260
7	Android Calculator	9.189
8	Twitter	8.645
9	Chrome Browser	8.524
10		
10	Yahoo Mail	3.287
9	ESPN SportsCenter	3.184
8	Google Search	1.984
7	Android Music	1.972
6	Pandora	1.779
5	Super-Bright LED Flashlight	1.667
4	UB Parking	1.507
3	NFL Mobile	1.437
2	Google+	1.270
1	Facebook Messenger	1.199

(c) **Apps Sorted by Foreground Energy Efficiency.**

Rank	App Name	Value
1	YouTube	18497.052
2	Candy Crush Saga	14051.369
3	Bank of America	12954.196
4	Dropbox	7063.746
5	Android Messaging	6555.140
6	Android Gallery	5773.902
7	Twitter	5610.394
8	Android Clock	5085.873
9	Yahoo Mail	5083.615
10		
10	NFL Mobile	1275.985
9	UB Parking	1071.529
8	Pandora	1049.971
7	Facebook Messenger	1012.536
6	Android News and Weather	990.386
5	Adobe Reader	985.680
4	Google+	898.589
3	Android Phone	748.077
2	Google Search	682.005
1	The Weather Channel	571.405

(d) **Apps Sorted by Content Energy Efficiency.**

Table 1: **Evaluating Components of a Value Measure.** PhoneLab data is used to weight overall app energy usage in a variety of different ways. Omitted results are caused by Android reporting energy consumption for non-apps such as the Android System.

very quickly, it may be more likely that it uses a "spinner" or other fancy UI elements that generate artificially high frame rates without delivering much information. The inability to distinguish between meaningless and meaningful video frame content is a significant weakness of this simple approach. YouTube and Candy Crush Saga both earn high marks, which is encouraging given that they are very different apps but also might be a result of overweighting screen refreshes. The Android Clock is also an unsurprising result, as it requires almost no energy to generate a relatively-large number of screen redraws in timer and stopwatch mode.

4.5 Survey Results and Discussion

To continue the evaluation of our simple content-based value measure, we prepared a survey for the 107 PhoneLab participants who contributed data to our experiment. Our goal was to determine if users would be more willing to remove inefficient apps, as defined using our content-based measure. As a baseline, we also asked users about the apps that consumed the most energy. We used each participants data to generate a custom survey containing questions about 9 apps: the 3 least efficient apps as computed by our content-based value measure, the 3 apps that used the most energy on their smartphone during the experiment, and 3 apps chosen at random. For each we asked them a simple question: "If it would improve your battery life, would you uninstall or stop using this app?" To compute an aggregate score for both the content-based and usage based measures, we give each measure 1 point for a "Yes", 0.5 points for a "Maybe" and 0 points for a "No". 47 participants completed the survey, and the results are shown in Figure 1. For each user, if the score of one measure is higher than the other, it is considered a "win" for the former.

Overall the results are inconclusive, with the content-delivery measure not clearly outperforming the straw-man usage measure at predicting which apps each user would be

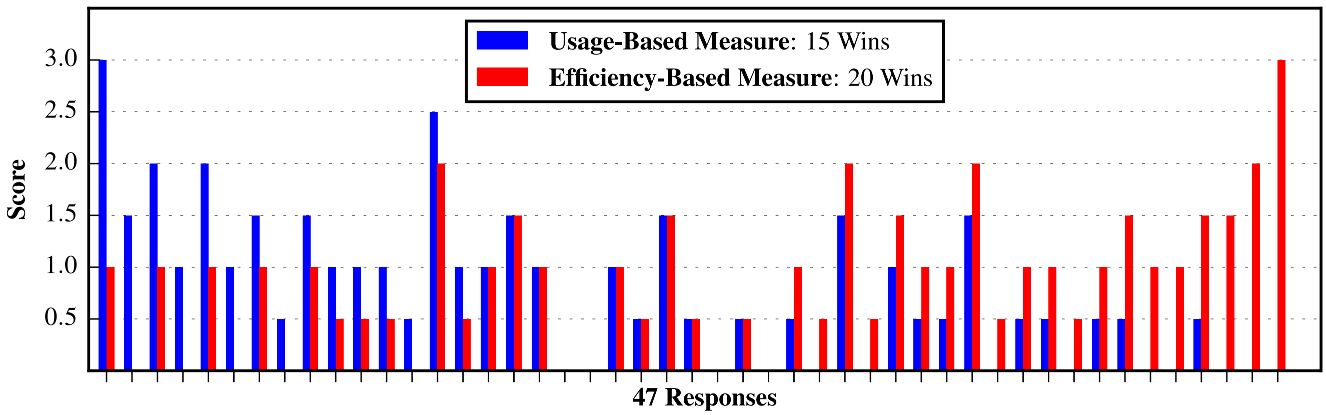

Figure 1: **Survey Results.** The height of each bar demonstrates how many of the suggested apps the user is willing to remove for better battery life, with suggestions based on overall usage or our new content-delivery efficiency measure. Our new measure does not convincingly out-perform the straw man.

willing to remove to save battery life. Given the crude nature of our metric, this is not particularly surprising, and can be interpreted as a sign that we need a more sophisticated value measure incorporating more of the potential inputs we have previously discussed. However, on one level the results are very encouraging: most users were willing to consider removing one or more apps if that app would improve their battery lifetime. Clearly, users are making this decision based on some idea of each app's value—the challenge is to replicate their choices using the information we have available to us.

5. CONCLUSIONS

To conclude, we have argued that our inability to estimate app value is a critical weakness that is threatening our successes at accurately estimating and attributing energy consumption. We have motivated the need for a value measure by describing the multiple ways in which it would aid in the management of energy and other resources on battery-powered smartphones. Using an energy consumption dataset collected on PhoneLab we have explored separately several potential inputs to a value measure and determined how they weight energy consumption. Finally, we have presented results from a failed effort to formulate an effective value measure. While this first attempt was unsuccessful, we hope to engage the mobile systems community in this effort so that more sophisticated and successful value measures can be developed.

Acknowledgments

Students and faculty working on estimating app value are supported by NSF awards 1205656 and 1423215. The authors thank the anonymous reviewers for their feedback.

6. REFERENCES

[1] Flinn, J., and Satyanarayanan, M. Energy-aware adaptation for mobile applications. SIGOPS Oper. Syst. Rev. 33, 5 (1999), 48–63.

[2] Fonseca, R., Dutta, P., Levis, P., and Stoica, I. Quanto: Tracking energy in networked embedded systems. In Proceedings of the USENIX Symposium on Operating Systems Design and Implementation (OSDI) (November 2008).

[3] Lorincz, K., rong Chen, B., Waterman, J., Werner-Allen, G., and Welsh, M. Resource aware programming in the pixie os. In Proc. 6th ACM Conference on Embedded Networked Sensor Systems (SenSys'08) (New York, NY, USA, 2008), ACM, pp. 211–224.

[4] Ma, X., Huang, P., Jin, X., Wang, P., Park, S., Shen, D., Zhou, Y., Saul, L. K., and Voelker, G. M. eDoctor: Automatically Diagnosing Abnormal Battery Drain Issues on Smartphones. In Proceedings of the 10th USENIX conference on Networked Systems Design and Implementation (Berkeley, CA, USA, 2013), NSDI'13, USENIX Association, pp. 57–70.

[5] Nandugudi, A., Maiti, A., Ki, T., Bulut, F., Demirbas, M., Kosar, T., Qiao, C., Ko, S. Y., and Challen, G. Phonelab: A large programmable smartphone testbed. In Proc. 1st International Workshop on Sensing and Big Data Mining (SenseMine 2013) (November 2013).

[6] Oliner, A. J., Iyer, A. P., Stoica, I., Lagerspetz, E., and Tarkoma, S. Carat: collaborative energy diagnosis for mobile devices. In SenSys (2013), C. Petrioli, L. P. Cox, and K. Whitehouse, Eds., ACM, p. 10.

[7] Pathak, A., Hu, Y. C., and Zhang, M. Where is the Energy Spent Inside My App?: Fine Grained Energy Accounting on Smartphones with Eprof. In Proceedings of the 7th ACM european conference on Computer Systems (New York, NY, USA, 2012), EuroSys '12, ACM, pp. 29–42.

[8] Pathak, A., Hu, Y. C., Zhang, M., Bahl, P., and Wang, Y.-M. Fine-Grained Power Modeling for Smartphones Using System Call Tracing. In Proceedings of the sixth conference on Computer systems (New York, NY, USA, 2011), EuroSys '11, ACM, pp. 153–168.

[9] Punzalan, R. Smartphone Battery Life a Critical Factor for Customer Satisfaction . http://www.brighthand.com/default.asp?newsID=18721.

[10] Roy, A., Rumble, S. M., Stutsman, R., Levis, P., Mazières, D., and Zeldovich, N. Energy management in mobile devices with the cinder operating system. In Proceedings of the Sixth Conference on Computer Systems (New York, NY, USA, 2011), EuroSys '11, ACM, pp. 139–152.

[11] Xu, F., Liu, Y., Li, Q., and Zhang, Y. V-edge: Fast self-constructive power modeling of smartphones based on battery voltage dynamics. In Proceedings of the 10th USENIX Conference on Networked Systems Design and Implementation (Berkeley, CA, USA, 2013), nsdi'13, USENIX Association, pp. 43–56.

[12] Yoon, C., Kim, D., Jung, W., Kang, C., and Cha, H. AppScope: Application Energy Metering Framework for Android Smartphones Using Kernel Activity Monitoring. In Proceedings of the 2012 USENIX conference on Annual Technical Conference (Berkeley, CA, USA, 2012), USENIX ATC'12, USENIX Association, pp. 36–36.

[13] Zeng, H., Ellis, C. S., Lebeck, A. R., and Vahdat, A. Currentcy: a unifying abstraction for expressing energy management policies. In Proceedings of the annual conference on USENIX Annual Technical Conference (Berkeley, CA, USA, 2003), USENIX Association, pp. 4–4.