Overcoming Energy and Reliability Challenges for IoT and Mobile Devices with Data Analytics

Sudeep Pasricha
Department of Electrical and Computer Engineering
Colorado State University, Fort Collins, Colorado, 80523
sudeep@colostate.edu

Abstract – A very large amount of data is produced by mobile and Internet-of-Thing (IoT) devices today. Increasing computational abilities and more sophisticated operating systems (OS) on these devices have allowed us to create applications that are able to leverage this data to deliver better services. But today's mobile and IoT solutions are heavily limited by low battery capacity and limited cooling capabilities. This motivates a search for new ways to optimize for energy-efficiency. Advanced data analytics and machine-learning techniques are becoming increasingly popular to analyze and extract meaning from Big Data. In this paper, we make the case for designing and deploying data analytics and learning mechanisms to improve energy-efficiency in IoT and mobile devices with minimal overheads. We focus on middleware for inserting energy-efficient data analytics-driven solutions and optimizations in a robust manner, without altering the OS or application code. We discuss several case studies of powerful and promising developments in deploying data analytics middleware for energy-efficient and robust execution of a variety of applications on commodity mobile devices.

Keywords: data analytics, middleware, mobile computing, Internet-of-Things (IoT), energy-efficiency, robustness

I. Introduction

We are well into the era of 'Big Data', with several zettabytes (ZB) of data being handled by datacenters annually, and the volume of this data expected to double every two years [1]. The increase in data over the past decade has been fueled by the proliferation of embedded Internet of Things (IoT) devices and smart mobile computing. For instance, a study from CISCO in 2016 [2] suggests that on an average 10.7 exabytes of mobile data traffic is offloaded each month from mobile phones to cloud datacenters. The study further forecasts that this value is set to increase 8-fold by 2021.

The generated Big Data can be structured (e.g., financial records), semi-structured (e.g., tweets), unstructured (e.g., audio, video), or real-time (e.g., monitoring logs). All of these types of data have the potential to provide invaluable insights, if organized and analyzed appropriately. The analyses of such large and diverse datasets is fast emerging as an indispensable tool for innovations in various domains such as healthcare, business process optimization, and social-network-based recommendations.

Unfortunately, the sheer volume of Big Data and its projected growth in the coming years will largely outpace foreseeable improvements in the cost, reliability, energy-efficiency, and performance of computing infrastructure such as mobile platforms and IoT devices. Thus, the design of these computing platforms requires significant innovations in the near future, to overcome increased energy costs due to the high computational power when processing and analyzing large datasets. As an example of the severity of the problem, consider the battery life of mobile and IoT devices. Although lithium-ion battery technology and capacity has improved over the years, it still cannot keep pace with the energy demands of today's mobile devices. Figure 1 shows the battery limits, thermal power limits, and desired power dissipation of mobile chipsets with respect to developments in mobile communication capabilities and chip technology. The gap between desired power (to meet growing performance needs) and battery limits is only getting larger, e.g., over a period of seven years, the processing capability of the Samsung Galaxy S series mobile device has grown by ~20×, whereas, the battery capacity has grown only by ~2× [3]. Battery life limitations today inevitably constrain performance for most applications on mobile devices.

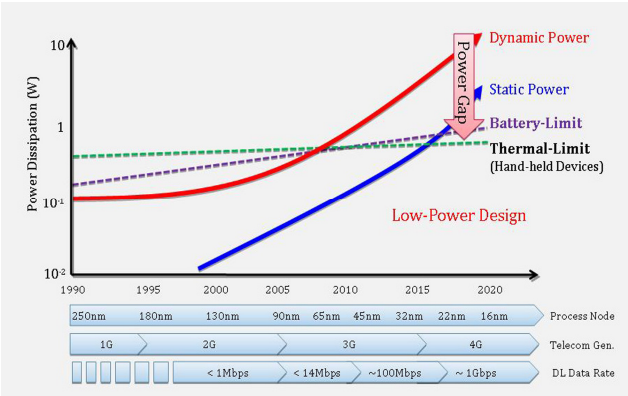


Figure 1: Trends for power dissipation as compared to battery and thermal limits in mobile devices since 1990 [4]

Fortunately, system designers that are today grappling with the challenge of efficiently supporting large datasets and data analytics workloads can themselves benefit from data analytics techniques to architect better computing platforms. For instance, in mobile computing, massive amount of data is typically collected about user-device interactions and usage. Can this data be analyzed to improve energy efficiency and robustness of applications executing on the mobile device? Even if it is possible to intelligently analyze and perform optimizations with the knowledge of such data,

where exactly in the software stack should the data analytics techniques be implemented: in the apps, OS, or elsewhere?

The answer to the latter question may lie with middleware, which is becoming an increasingly important component in modern mobile and IoT device software stacks [5]. Middleware is intended to provide services or functionalities to the application developer that are not already a part of the OS. A salient feature of middleware is the abstraction level that it can create for the application layer. An application developer does not need to be aware of how different modules come together in the middleware for a new service to work. Thus middleware has been growing in popularity, and has especially found widespread application in distributed and cloud based applications. An example is Microsoft Azure [6], that is a growing collection of integrated services (applications) and platforms (operating system) combined using middleware to deliver numerous services-on-demand in manner that is personalized per user.

In this paper, we explore the design and implementation of intelligent data-analytics based middleware for mobile and IoT devices that can assist with improving the energy-efficiency and robustness of these devices. The abstraction advantage and modularity of middleware allows it to be a very good solution to swiftly create and deploy services that enhance the existing features of an OS. Several case studies of powerful and promising developments in prototyping data analytics middleware for energy-efficient and robust execution of a variety of applications on commodity mobile computing devices are discussed in this paper.

The rest of this paper is organized as follows. Section 2 discusses middleware to enable mobile-to-cloud offloading. Section 3 discusses middleware for user-interaction aware execution of applications on mobile devices. Section 4 presents middleware that captures spatio-temporal context for various optimizations. Section 5 explores middleware for mobile indoor localization.

II. Mobile-To-Cloud Offloading

The collection and processing of data on smartphones can significantly hamper the battery lifetime of the device. A promising solution that is being considered to support high end mobile data processing applications is to offload mobile computations to the cloud [7]-[9]. Offloading is an opportunistic process that relies on cloud servers to execute the functionality of an application that typically runs on a mobile device. In many cases, such opportunistic offloading can not only improve energy-efficiency but also makes computation performance more robust.

Kumar et al. [8] presented a mathematical analysis of offloading. Broadly, the energy saved by computation offloading depends on the amount of computation to be performed (C), the amount of data to be transmitted (D), and the wireless network bandwidth (B). If (D/C) is low, then it was claimed that offloading can save energy. Our experiments have shown that this is a simplistic view of the problem, e.g., energy-efficiency is also highly effected by the type of wireless interface being used for the transmission. Cuervo et.al [7] proposed a framework called MAUI, based on code annotations to specify which methods from a software class can be offloaded to the cloud. Annotations are introduced in the source code by the developer during the development phase. At runtime, methods are identified by the MAUI profiler, which performs the offloading of the methods, if the bandwidth of the network and data transfer conditions are ideal. However, this annotation method puts an extra burden on the already complex mobile application development process. A better approach would be to have a middleware that is able to automatically make intelligent offloading decisions on the fly, without manual annotations.

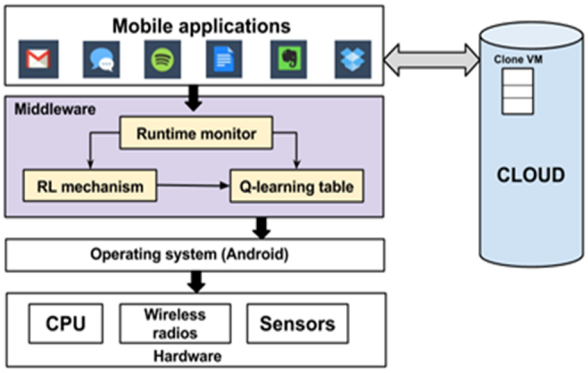


Figure 2. Reinforcement Learning (RL) based middleware framework for efficient application offloading from mobile device to the cloud [10]

In [10], we proposed a novel middleware framework for mobile devices that utilized various sources of data such as the app communication/computation intensiveness, type and status of available wireless networks, and capabilities of cloud servers, to dynamically make decisions on when and how to offload an application from the mobile device to the cloud, with the goal of improving energy-efficiency and performance robustness. Figure 2 shows an overview of our data analytics decision engine on the mobile device that works together with a clone virtual machine (VM) of the mobile software environment to execute apps on cloud servers. The middleware was deployed on the Android OS, and ran in the background as an Android service. The framework utilized an unsupervised Q-learning machine learning technique that analyzed the data for the app type, network type, network conditions, and cloud capabilities to select the optimal network type and decide when and what to offload.

Figure 3 shows an example of how the Android based torrent app Flud [11] can benefit from our middleware-based offloading framework. In this experiment, a cloud based service (Amazon Web Services EC2 instance) first downloads and aggregates the file to be received through torrent, and then the smartphone downloads the file in a single process. The experiment was conducted on an LG G3 Android smartphone. From the bars in the figure, it can be observed that different network types, the state of the network, and data transfer sizes result in varying improvements in energy and response time. For instance, 4G performs slightly better

than 3G in terms of energy consumption for higher data sizes (45-85 MBs), but for smaller data sizes 3G is more efficient. The colored lines in Figure 3 indicate the performance of our middleware framework (green line) and a framework based on fuzzy logic for making offloading decisions (red line) [9]. In all cases, it can be observed that our framework is able to provide better offloading performance and greater energy efficiency. This is because our framework employs a more sophisticated and powerful data analytics-based learning algorithm and considers many more variables related to device and network context when making decisions, than prior work. Our experiments with real smartphones showed savings of up to 30% in battery life with up to 25% better response time when using our middleware framework compared to a state-of-the-art fuzzy logic based offloading approach [9]. For certain applications, e.g., voice recognition, offloading also improved recognition robustness (accuracy) by approx. 10%.

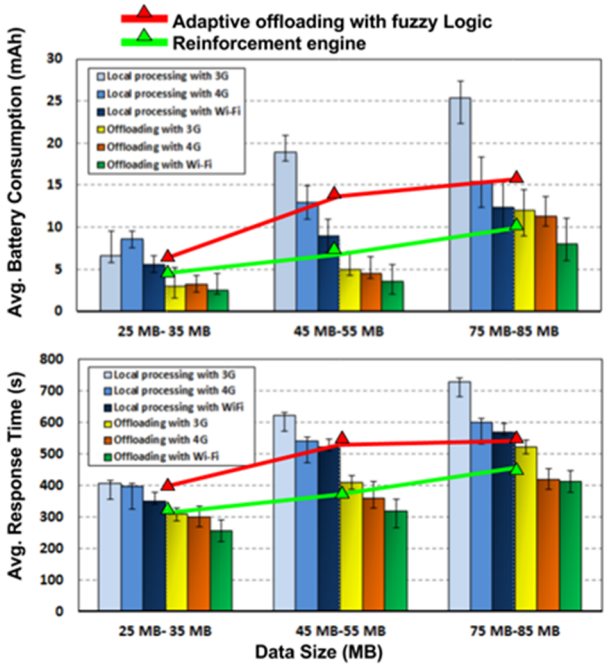


Figure 3: Average battery consumption and response time on a mobile device for a torrent file download application [10]

III. User-Interaction Aware Optimizations

Mobile devices today are seen as a personal tool and their typical usage can vary across different users. OS- and hardware-driven energy-optimization techniques, such as CPU dynamic voltage and frequency scaling (DVFS), are not smart enough to make decisions based on the user's behavior. To enable more aggressive energy optimizations in mobile devices, we developed a novel application- and user-interaction aware energy management middleware framework (AURA) for mobile devices [12], [13]. AURA takes advantage of user idle time between interaction events of the foreground application to optimize CPU and backlight energy consumption. Most interestingly, AURA is able to adapt to changing behavior and learn from the individual user over time, to achieve longer battery lifetime without any user intervention.

To create an energy-efficient middleware solution, it is important to first identify the components of the mobile device that are major contributors towards battery life. Our preliminary analysis revealed that the display, the processing (CPU/GPU) subsystem, and the various wireless radios (e.g., Wi-Fi, GPS, 4G/LTE) have a significant impact on battery life. Any framework that aims to optimize energy-efficiency must address the energy inefficiencies in these components. Our AURA framework [12], [13] was one of the first to reduce energy costs for both the display and the processing subsystems in an integrated manner. The framework consists of an app-aware and user-aware energy optimization middleware that uses powerful machine learning techniques on user-device interaction data. More specifically, AURA includes a runtime monitor that captures data related to user-specific and app-specific interaction slack to reduce energy costs.

Interaction slack refers to the sum of the unused times between when a user first perceives a change on the display (perceptual slack) due to a previous interaction (e.g., a button on the screen changes color), then comprehends what the response "actually" represents (cognitive slack), and finally interacts with application again by touching the screen using his/her fingers (motor slack). By predicting this interaction slack interval on a per-app and per-user basis, AURA opportunistically reduce CPU voltage/frequency at the start of the interval and then increase the voltage/frequency just before the interval ends, to save energy without impacting user quality of service (QoS).

AURA's middleware was prototyped as a service that constantly runs in the background and creates an automated control system for CPU voltage/frequency scaling and display backlight modifications. The middleware consist of three main components: a runtime monitor, a Bayesian app classifier, and a power manager. The runtime monitor checks if the current foreground app has an entry in an 'interaction database', and if so then the interaction data (standard deviation and mean of a user's observed slack values from previous interactions) in it can be used for slack prediction. If a database entry does not exist, the middleware creates a new entry and starts collecting interaction statistics. A Bayesian classifier is then used to classify the interaction profile for the app using the collected data. Bayesian learning is a form of supervised machine learning that involves using evidence or observations along with prior outcome probabilities to calculate the probability of an outcome. The power manager runs a MDP (Markov Decision Process) to classify the apps. All apps were classified into seven categories ranging from very-low-interaction to very-high-interaction. The class of an app decides how to opportunistically decrease CPU voltage/frequency in between slack intervals. MDPs are discrete time stochastic control processes that are widely used as decision-making

models for systems in which outcomes are partly random and partly controlled.

In [12], we explored two derivatives of the (normal) MDP to dynamically adapt to real-time user-interaction during each invocation of an application. The E-ADAPT variant is event-driven and uses the most recent window of events to predict future interaction events whereas T-ADAPT makes use of a moving average window of a predetermined size, to dynamically track and predict user interaction events in a temporal context. In [13], we prototyped a new Q-learning based power manager. Q-Learning does not require a model of the environment and has the advantage that the next state probability distributions that are used in MDPs are not required. Using the Android services based middleware approach allowed us to the rapidly develop, update, deploy, and test new versions of AURA.

The AURA middleware also exploits the idea of change blindness [14] as identified by research into human psychology and perception. Change blindness refers to the inability of humans to notice large changes in their environments, especially if changes occur in small increments. Multiple studies have shown this as a limitation of human perception, e.g., a majority of observers in one study failed to observe when a building in a photograph gradually disappeared over the course of 12 seconds. We used a similar approach by gradually reducing screen brightness over time using user-device and app-specific interaction data. In doing so, the power manager in AURA is able to achieve higher energy-efficiency without any noticeable loss in QoS.

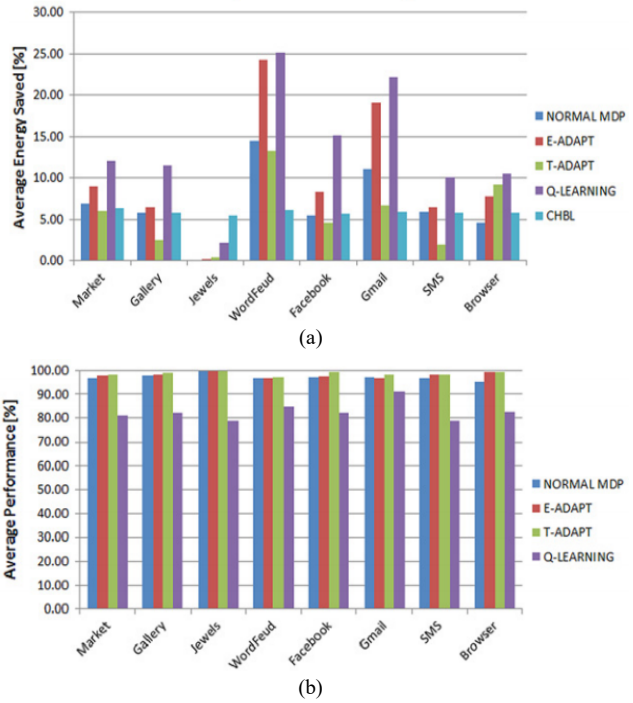


Figure 4: Real user study results on HTC Dream mobile device [13]

We deployed our AURA middleware framework on several smartphones such as the HTC Dream and Google Nexus One. Figure 4 shows the results for energy savings and interaction slack prediction on the HTC Dream smartphone across various common mobile applications, averaged for several real users. In addition to the four variants of AURA (with power managers based on NORMAL-MDP, E-ADAPT, T-ADAPT, and Q-LEARNING) we compare against Shye et al.'s algorithm, CHBL [15], which was the best known algorithm for energy savings on mobile devices. It can be seen from Figure 4(a) that our user-aware and application-aware algorithms (particularly Q-LEARNING) offer higher energy savings than CHBL because unlike CHBL they can dynamically adapt to the user-interaction patterns and take full advantage of user idle time. Figure 4(b) shows the average successful prediction rates for the real user interaction patterns. CHBL was not included in the results because it does not contain defined states or prediction mechanisms, making determining mispredictions impossible. The figures show high successful prediction rates with AURA that result in high QoS for users. Our extensive experiments indicated 17% energy savings on average for AURA compared to the baseline Android device manager and approximately 2.5× more energy savings over the best known prior work (CHBL [15]) on mobile CPU and display energy optimization.

IV. Spatiotemporal Context-Aware Optimization

Today mobile devices come with a variety of wireless interfaces such as GPS, Wi-Fi and 3G/4G. Our experimental analysis on various smartphones [16] indicated that even when 3G/4G, Wi-Fi, and GPS interfaces are all enabled and idle, they account for more than 25% of total system power dissipation. Furthermore, when only one of the interfaces is active, the other idle interfaces still consume a non-negligible amount of power. This is the motivation behind efforts to enable intelligent management of such wireless interfaces. It is important to note that activation of wireless interfaces, for location or data, is directly correlated with the context of the device itself, e.g., the type of application running, device motion, Wi-Fi availability, time of day, location, etc. This observation opens up an opportunity to realize a context-aware solution that is able to more efficiently manage wireless interfaces without human intervention.

Our middleware framework in [17], [18] represents one of the first efforts towards seamless wireless interface energy management in mobile devices. The first step in our approach is to collect and learn from the contextual data of the device, the user, as well as the state of wireless interfaces. Our framework is able to transparently capture contextual data attributes such as temporal use data (e.g., day of week and time), spatial environment data (e.g., ambient light, Wi-Fi RSSI, 3G/4G signal strength), and device state (e.g., battery status, CPU utilization). To prune the massive amount of data captured, we employed Principal Component Analysis (PCA), a form of dimensionality reduction, by projecting the captured data from various sources onto a

fewer number of optimally selected eigenvectors, effectively reducing the attribute space to the (limited) number of eigenvectors, to enable efficient prediction on resource-constrained mobile devices.

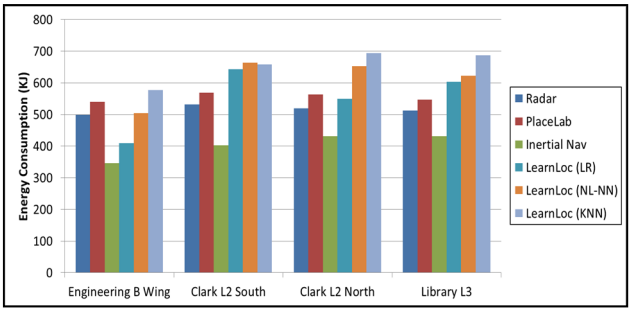
We then explored the use of five different classes of machine learning algorithms to learn from this contextual data. The algorithms we considered included LDA (Linear Discriminant Analysis), LLR (Linear Logistic Regression), NN (Shallow and Deep Neural Networks), KNN (K-Nearest Neighbor), and SVM (Support Vector Machines). These algorithms allowed us to predict user data/location usage requirements (e.g., is data transfer needed? is coarse-grained location needed? is fine-grained location needed?) based on the spatiotemporal user and device context data collected.

We found that SVM and deeper NNs (with more hidden layers) resulted in the highest context prediction accuracy (85 – 99%). KNN, LLR, and LDA also performed fairly well, with prediction accuracies in the range of 60 – 90 %. The highest energy savings were achieved with LDA and LLR. However, these higher savings come at the cost of degraded user satisfaction (highlighted by their lower context prediction accuracy). SVM and KNN overall performed fairly well in terms of both prediction accuracy and energy savings potential, as did the NN approach. But KNN's run time on a mobile device is several orders of magnitude larger than any of the other algorithms, because all computations are deferred until its classification phase. Therefore, although KNN is as good as or better than the SVM and NN based approaches in terms of energy savings and prediction accuracy, it is not the most practical for deployment on mobile devices. Our SVM based middleware prototype provides good accuracy, good energy savings, and demonstrates the best adaptation to various unique user usage patterns, while maintaining a low implementation overhead on mobile devices. We showed approximately 85% energy savings with SVM for minimally active users

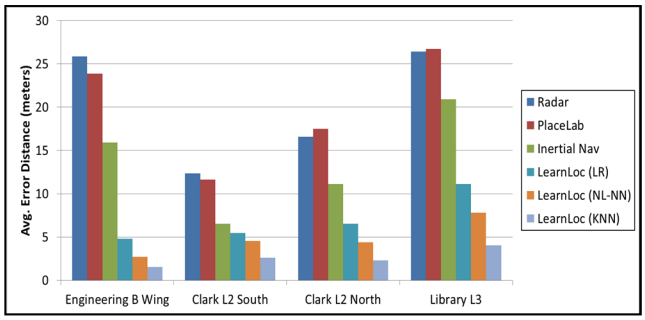
V. Mobile Indoor Localization

Location tracking has found various applications outdoors. One can not only use GPS based services for navigation purposes, but companies such as Google have been providing users location based services such as locating the best places to eat in their vicinity, local news, local weather, reminders to get an item when near a grocery store [19], etc. The outdoor location based services available today are extremely helpful, yet, in most cases they are limited once a user moves indoors. For instance, in the previously suggested example of reminders when near grocery stores, the application can only remind a user to buy the item from the store but is unable to provide any guidance on how to locate that item within the store. There are several other use cases that remain unrealized, such as being reminded to go to a certain store within a large indoor mall, notifications to the user when they are close to specific items/aisles in a store, or navigation help to reach specific rooms in a large building. These and many other examples make a strong case for the creation of indoor localization solutions on devices that most people carry with them everywhere: smartphones.

Indoor localization is a challenge that cannot be resolved through a conventional outdoor solution such as GPS. This is because GPS signals are weak and ineffective in indoor environments, and the wireless signal-based infrastructure for indoor localization is diverse, prone to interference, and often entirely non-existent [20]. A possible approach to overcome this challenge is fingerprinting, where the goal is to use data captured through smartphone radio interfaces and sensors to estimate the location of the user indoors (inside of buildings, caves, etc.) in real time. However, continuous monitoring of radio and sensor data drains battery life, thus indoor localization solutions must be energy-aware.



(a)



(b)

Figure 5: Comparison of indoor localization techniques [21].

We devised the LearnLoc middleware framework for mobile devices in [21] to improve indoor localization reliability (accuracy) and energy efficiency in a variety of indoor environments. We capture Wi-Fi "fingerprint" data (signal strength, signal type, access point ID) in a continually updated database for indoor locales together with inertial sensing readings (accelerometer, gyroscope, and magnetometer sensor data) on mobile devices. This data is then analyzed and parsed with machine learning techniques to estimate location in real time. The framework is implemented as a middleware service on mobile devices to provide indoor location based updates and suggestions in a non-intrusive and energy-efficient manner. The most important feature of our approach is that it does not require any additional hardware setup indoors, as Wi-Fi access points have become increasingly common in indoor public spaces which brings down the cost of realizing indoor localization.

In [21], we prototyped and compared three variants of the LearnLoc framework that use Linear Regression (LR), non-linear neural networks (NL-NN) and K-nearest neighbor (KNN) techniques. The use of smart machine learning techniques helps LearnLoc significantly improve prediction accuracy and overcome noisy data readings compared to prior work. Figure 5 shows a comparison of energy consumption and localization accuracy of the three variants of the LearnLoc framework with well-known techniques from prior work (Radar [22], PlaceLab [23], Inertial_Nav [24]) along four indoor paths of 110m-140m lengths in various buildings at Colorado State University. We observed that PlaceLab and Radar consumed much less energy but that comes at a price of accuracy. The experimental results suggest that KNN delivers the most accurate location estimates, but also consumes the most energy. The LR and NL-NN variants perform considerably well overall. It is also important to note that Wi-Fi scan interval also plays a significant role in accuracy and energy-efficiency. The lowest Wi-Fi scan interval (1 second) delivers the best result, but a balance between energy consumption and accuracy can be achieved through the selection of a balanced scan interval. By prototyping a middleware framework that enables trade-offs between energy and accuracy, our work has brought viable indoor localization solutions that can be implemented on commodity mobile devices closer to reality.

VI. Conclusions

The rise of the data-driven science paradigm, in which massive amounts of data are produced and processed by mobile and IoT devices on a very strict power budget, requires new solutions to sustainably use these devices. Energy-efficient and robust data analytics approaches can help make the most out of available hardware resources. In this paper, we have shown, through various real-world case studies, that the flexibility and capabilities provided by such smart data analytics and machine learning middleware solutions can make a significant impact towards enhancing energy-efficiency, reliability, and performance robustness in a variety of mobile and IoT devices.

Acknowledgements

This work was supported by grants from the National Science Foundation (ECCS- 1646562, CCF-1252500).

References

[1] 'Mobile data will skyrocket 700% by 2021', 2017 [online] Available: http://www.businessinsider.com/mobile-data-will-skyrocket-700-by-2021-2017-2

[2] Cisco Global Cloud Index: Forecast and Methodology, 2015–2020, White paper, [Accessed: Oct 2017]

[3] 'Samsung Galaxy S specifications', [Online], Available: https://www.phonearena.com/phones/Samsung-Galaxy-S_id4522

[4] 'System level Power Budgeting', 2017 [Online] Available: http://chipdesignmag.com/sld/blog/2014/03/12/system-level-power-budgeting/ [Accessed: Oct 2017]

[5] S. Tiku, S. Pasricha, "Energy-Efficient and Robust Middleware Prototyping for Smart Mobile Computing," IEEE International Symposium on Rapid System Prototyping (RSP), Oct 2017.

[6] 'Microsoft Azure, [online], Available: https://azure.microsoft.com/en-us/overview/what-is-middleware/

[7] E. Cuervo, A. Balasubramanian, D. K. Cho, A.Wolman, S. Saroiu, R. Chandra, and P. Bahl, "Maui: making smartphones last longer with code offload," in Proc. ACM Mobisys, 2010.

[8] K. Kumar and Y.-H. Lu, "Cloud computing for mobile users: Can offloading computation save energy?" Computer, vol. 43, 2010.

[9] H. R. Flores and S. Srirama, "Adaptive code offloading for mobile cloud applications: Exploiting fuzzy sets and evidence-based learning," Proc. ACM Mobisys, 2013.

[10] A. Khune, S. Pasricha, "Mobile Network-Aware Middleware Framework for Energy Efficient Cloud Offloading of Smartphone Applications", to appear, IEEE Consumer Electronics, 2017.

[11] 'Fuld – Torrent Downloader app', 2016, [Online]. Available: https://play.google.com/store/apps/details?id=com.delphicoder.flud&hl=en, [Accessed: Oct 2017].

[12] B. Donohoo, C. Ohlsen, S. Pasricha, "AURA: An Application and User Interaction Aware Middleware Framework for Energy Optimization in Mobile Devices", IEEE International Conference on Computer Design (ICCD 2011), Oct. 2011.

[13] B. Donohoo, C. Ohlsen, S. Pasricha, "A Middleware Framework for Application-aware and User-specific Energy Optimization in Smart Mobile Devices", Journal of Pervasive and Mobile Computing, vol. 20, pp. 47-63, Jul 2015.

[14] D.J. Simons, R.A. Rensink, "Change blindness: Past, present, and future", Trends Cogn. Sci. 9 (1) (2005).

[15] A. Shye, B. Scholbrock and G. Memik, "Into the wild: Studying real user activity patterns to guide power optimizations for mobile architectures," 42nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), pp. 168-178, 2009.

[16] "Google Nexus One Tech Specs" [Online]. Available: http://www.htc.com/us/support/nexus-one-google/tech-specs

[17] B. Donohoo, C. Ohlsen, S. Pasricha, C. Anderson, "Exploiting Spatiotemporal and Device Contexts for Energy-Efficient Mobile Embedded Systems", IEEE/ACM Design Automation Conference (DAC 2012), Jul. 2012.

[18] B. Donohoo, C. Ohlsen, S. Pasricha, C. Anderson, Y. Xiang, "Context-Aware Energy Enhancements for Smart Mobile Devices", IEEE Trans. on Mobile Computing (TMC), 13(8):1720-1732, 2014.

[19] 'Guide to Google Cards, 2017 [Online] Available: https://www.androidcentral.com/ultimate-guide-google-now-cards

[20] C. Langlois, S. Tiku, S. Pasricha, "Indoor localization with smartphones", IEEE Consumer Electronics, Oct 2017.

[21] S. Pasricha, V. Ugave, Q. Han and C. Anderson, "LearnLoc: A Framework for Smart Indoor Localization with Embedded Mobile Devices," ACM/IEEE CODES+ISSS, Oct 2015.

[22] P. Bahl, and V. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," IEEE INFOCOM. 2000.

[23] A. LaMarca, et al., "Place Lab: Device Positioning Using Radio Beacons in the Wild," Proc. PERCOM, 2005, pp. 116-133.

[24] J. Á. B. Link, P. Smith, N. Viol and K. Wehrle, "FootPath: Accurate map-based indoor navigation using smartphones," IPIN 2011.