This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MDAT.2020.2976669, IEEE Design and Test

A Survey on Energy Management for Mobile and IoT Devices

Sudeep Pasricha (sudeep@colostate.edu), Raid Ayoub (raid.ayoub@intel.com), Michael Kishinevsky (michael.kishinevsky@intel.com), Sumit K. Mandal (skmandal@asu.edu), Umit Y. Ogras (umit@asu.edu)

Abstract: Smartphones, wearables, and Internet of Things (IoT) devices have already revolutionized many aspects of our everyday lives, enabling ubiquitously connected machine-to-machine, machine-to-human, and human-to-human communication towards the realization of smarter environments in our homes and across the domains of medical care, transportation, energy grids, industry automation, and defense. This article surveys the landscape of energy management solutions for mobile and IoT devices, to maximize performance and quality-of-service within the varying resource-constraints of these devices.

Keywords: smartphones, wearables, IoT, energy-efficiency, mobile computing, energy harvesting

1 Introduction

We are well into the era of explosive growth for Internet of Things (IoT) devices, with almost 30 billion deployed worldwide today, and the number expected to grow to more than 75 billion by 2025 (about 10 devices for every human on this planet). IoT represents a growing network of heterogeneous devices, combining commercial, industrial, residential and cloud-edge computing domains. These devices range from low-power sensors with limited capabilities to multi-core platforms on the high-end. Smart IoT devices that are part of this network take many forms: industrial IoT devices controlling and analyzing manufacturing lines, cameras, watches, speakers, thermostats, drones, lights, sprinkler controllers, door locks, retail kiosks, etc.; all with the defining characteristic of having an IP address for Internet connectivity, allowing communication and data exchange with other devices and users.

Beyond IoT, most people are connected to the Internet via mobile devices, such as smartphones and tablets. There are more than 5 billion smartphones in use around the world today. It is projected that the user base just for this class of mobile devices will grow to 6 billion by 2025, covering 71% of the world's population.

Together, IoT and mobile devices are enabling a connected future that promises savings of time and money with better automation and control in industry and our everyday activities, as well as other benefits such as better health care via remote monitoring, reduced electricity usage in smart homes and offices, efficient fuel usage in smart and increasingly autonomous vehicles, and more potent conservation efforts. e.g., monitoring-driven initiatives to enhance air and water quality, etc.

All mobile devices and most IoT devices are portable, requiring a battery to operate. Typically, such devices have relatively small form factors, which limits the size of the battery that can be used in these devices. Li-ion rechargeable batteries are the most widely used batteries in these devices. However, the energy density of this battery technology has improved only minimally over the past few decades, and dramatically better alternatives have yet to be found. The resulting limited energy available from these batteries in turn limits the capabilities of components that can be used in these devices, such as sensors, processors, wireless interfaces, memories, and displays. This is a major challenge, especially given the need to run ever-more demanding applications on IoT and mobile devices, e.g., deep learning inference, augmented and virtual reality, and high definition video processing.

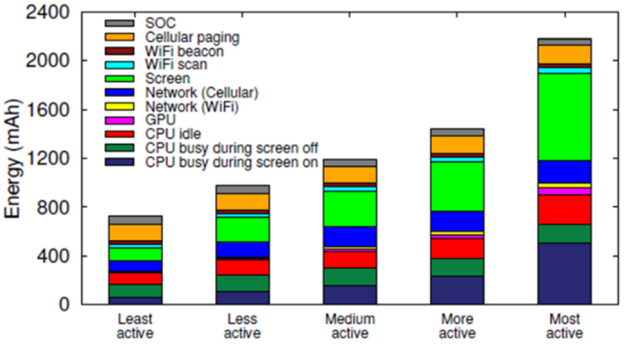

Fig 1. Average daily energy drain breakdown of 5 groups of 1520 users using the Samsung Galaxy S3 and S4 smartphones [1].

Figure 1 shows a breakdown of the average energy usage across 1520 users of the Samsung Galaxy S3 and S4 mobile devices [1]. The users are divided into five groups based on their activity levels. It can be observed that the display (screen), processors (CPUs, GPUs), wireless network radios (Wi-Fi, cellular) and the system-on-chip (SoC) consume varying amounts of energy on average. There is a strong motivation to minimize energy (and power) across all of these components, to (1) allow mobile and IoT devices to last longer on a single battery charge (i.e., increased device uptime or battery lifetime); (2) enable more sophisticated components such as faster CPUs, GPUs, and neural processing units (NPUs) to be selected if their energy and power footprint can be intelligently managed, (3) prevent thermal emergencies which can cause undesirable performance throttling and component failures; and (4) use cheaper and less bulky cooling and power management integrated circuit sub-components, to achieve even smaller form factors and cost savings.

There are many efforts today that are actively attempting to minimize the energy and power footprints of high-end IoT and mobile devices, as well as low-end IoT devices.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MDAT.2020.2976669, IEEE Design and Test

Widely used mobile and IoT CPUs (e.g., ARM's Cortex family [2]) and GPUs (e.g., Qualcomm's Adreno [3]) now include a large number of low-power and deep sleep states that can be quickly transitioned into and out of, to save energy and reduce power. Memory technologies such as LP-DDR5 [4] and Macronix low-power Flash [5] allow support for low power main memory and secondary storage operation. New wireless communication standards and protocols are being designed for low power wireless communication, e.g., IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN) [6], long-range wide-area network (LoRaWAN) [7] for long-range low-power communication, IPv6 routing protocol for low-power and lossy networks (RPL) [8], Bluetooth low-energy (BLE) [9], and LTE Release 12 [10] which provides a power saving mode and lower overhead signaling procedure to improve energy efficiency. Low overhead operating systems such as Contiki [11], TinyOS [12], FreeRTOS [13], and Zephyr [14] provide lightweight software stacks for resource-constrained IoT devices. Even high-end IoT and mobile devices utilize operating systems such as the Android OS [15], which are increasingly optimized for working with low power modes to extend battery lifetime.

Despite the promising developments highlighted above, there is still a huge design space for further energy optimizations, with opportunities to even more aggressively reduce energy in IoT and mobile devices. This article surveys the landscape of such approaches to reduce energy in IoT and mobile devices. Sections 2 through 5 review strategies to minimize energy in hardware sub-components, including processors, displays, wireless radios, and memory/storage. Section 6 describes software-centric energy optimizations. Section 7 presents cloud offloading strategies for saving energy. Section 8 discusses battery-aware aproaches. Section 9 describes user-aware dynamic optimizations. We conclude with Section 10, which discusses the remaining and emerging opportunities and challenges for energy optimization across IoT and mobile devices.

2 Processing Optimizations

State-of-the-art mobile platforms integrate multiple general purpose CPUs, graphics processing units (GPUs), and specialized processing elements (PEs), such as audio, video, and security engines [16], [17]. While these processing units improve user experience, they also increase power consumption, especially, when they are used heavily. For example, GPU power consumption dominates the SoC power while running graphics intensive games [18]. Thus, platform-level power management of all processing units in mobile platforms is a key research problem [19].

Core configurations and the operating frequency of PEs can be controlled at runtime. For example, Snapdragon and Exynos SoCs enable controlling the frequency of different CPU clusters independently and provide over ten voltage-frequency levels for each cluster [16], [20]. Power management governors embedded in operating systems, such as *powersave*, *performance*, and *interactive governors*, support control of power states dynamically at runtime [21]. These governors implement simple algorithms to control the PE frequencies as a function of their utilization. Hence, they leave much room for dynamic optimizations. Due to the emergence of millions of mobile applications, power management of mobile platforms has become necessary, but remains a difficult problem.

In the past decade, multiple researchers have addressed the power management problem for mobile platforms [22]-[24]. Most of the modern-day mobile platforms integrate CPU and GPU within a single system. Power management technique for these systems optimize the performance, measured in frames per second (FPS) under power consumption and thermal constraints [18], [25]. A task allocation strategy for heterogeneous mobile systems was presented in [26]. This approach, called SPARTA, first profiles the application behavior at runtime. Then, a heuristic algorithm prioritizes and allocates tasks at runtime. Another runtime task allocation approach for heterogeneous mobile system was proposed in [27]. This approach chooses the optimal design point from a set of design points while maintaining required performance. However, the aforementioned power management techniques primarily rely on heuristic algorithms that do not guarantee optimality for a given application. Qiu et al. [28] modeled the mobile system as stochastic service request process and formulate dynamic power management as a policy optimization problem. The authors solve this problem through a policy iteration algorithm and evaluate it using an event-driven simulator.

Machine Learning

A new class of dynamic management algorithms have emerged with the advent of machine learning (ML) techniques. A number of recent techniques construct multiple ML-based policies offline [29]-[31]. These techniques characterize applications during execution and choose a suitable pre-existing power management policy. The major drawback of this approach is the inability to capture dynamic workload variations [32]. Gupta et al. [33] proposed a phase level instrumentation technique to collect workload statistics at runtime. Specifically, the workload is divided into snippets and performance application programming interface (PAPI) calls are inserted between each snippet. Data collected for each snippet is then used to control the power states of PEs.

Reinforcement Learning (RL) is a widely used machine learning technique which enables online learning [34]. RL is a model free technique where the policy takes an action and receives a reward from the environment based on the action. Then, the policy is updated using the reward. Several researchers have proposed power management technique using Q-learning [35]-[38]. The methodologies proposed in [35], [36] use a table to store the Q-values for state-action pairs. Since the system states in standard mobile processors are usually continuous, they are divided into discrete bins and stored in the Q-table. The efficiency of the approach depends on the number of discrete bins in the Q-table. If high accuracy of the policy is intended, then the size of the Q-table can become very large. Such large Q-table sizes in turn require additional memory within the platform and increase the execution time. To address the limitations of the Q-table based approach, deep Q-learning

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MDAT.2020.2976669, IEEE Design and Test

Table 1: Methodology and the scope of different power management approaches for mobile platforms.

Reference	Methodology	Evaluation Platform	Single-threaded	Multi-threaded	Gaming
[28]	Policy Iteration	Event-driven simulation	✓	×	×
[25]	Heuristic	Odroid XU+E	×	×	✓
[26], [27]	Heuristic	Odroid XU3	✓	✓	×
[43]	Control-theoretic	Baytrail SoC	×	×	✓
[29]	Multivariate Linear Regression	Odroid XU3	✓	✓	×
[33]	Logistic Regression	Odroid XU3	✓	✓	×
[36], [37]	Reinforcement Learning	Simulation	✓	×	×
[38]	Reinforcement Learning	Odroid XU3	✓	✓	×
[41]	Imitation Learning	Gem5	✓	✓	×
[42]	Imitation Learning	Odroid XU3	✓	✓	✓

based power management policies approximate Q-tables using a deep neural network and experience replay buffers [37], [38].

RL-based power management policies have two critical drawbacks. *First,* RL takes a significant number of iterations to converge to the optimal policy. *Second,* the efficiency of RL depends on the design of the reward function. It is hard to design a single reward function that will produce a good power management policy across different platforms. Imitation Learning (IL) is an effective ML technique suitable for sequential decision-making problems [39]. IL techniques construct a policy by using an Oracle that captures the optimal behavior. Since an exact-IL approach can suffer from error propagation, data aggregation algorithms are used to construct the policy [40]. The first application of IL techniques to dynamic power management was presented in [41]. However, this technique is only applicable to homogeneous processors. A recent technique develops a DPM policy for heterogeneous mobile platforms by constructing an Oracle using dynamic programming [42]. The IL-based policy was shown to achieve significant improvement in energy consumption with respect to default governors in mobile platforms.

Apart from CPUs, different modeling and management techniques are proposed for GPUs [43]-[45] as well as interconnects and caches [46], [47]. Machine learning-based models are used to predict GPU performance in [43]. Similarly, Deitrich et al. [44] proposed an autoregressive offline model to estimate GPU performance. However, these models use the same features for all applications. This drawback is addressed by using GPU performance models that adapt to the workload at runtime [45]. In this methodology, features are selected offline and model coefficients are learned online through a Recursive Least Square (RLS) technique. State-of-the-art platforms enable controlling the frequency of on-chip interconnects and caches (i.e., uncore) [48]. Recent techniques predict workload characteristics and control the uncore frequency to minimize power consumption with negligible performance loss [46], [47]. A summary of methodology and scope of different power management techniques for mobile platforms is provided in Table 1. This table shows which power management strategy is suited for different kind of workloads (such as single- and multi-threaded applications, gaming applications), as well as the evaluation platform. Lastly, a comprehensive and systematic review of on-chip resource management techniques can be found in [49].

3 Display Optimizations

Display screens on smartphones, smartwatches, and tablets tend to consume a major portion of overall energy and also power at any given time. Many IoT devices (e.g., smart thermostats, wireless weather stations) also rely on displays. There are three popular display technologies that are widely used in mobile and IoT devices: (1) Liquid crystal display (LCD); (2) Organic light emitting diode (OLED) display; and (3) Active matrix electrophoretic display (AMEPD) which is more widely known as e-paper.

LCD Displays

LCD screens are the oldest display technology, first released for a mobile device in 1992, as part of the Simon Personal Communicator which featured a black-and-white 160 x 293 LCD touchscreen measuring 4.5 inches by 1.4 inches. Around the early 2000s, companies such as Nokia and Sony released phones with color LCDs, offering 256 colors. LCD displays rely on a white backlight (or sidelight) that passes light through layers of polarizing filters (which make all the light oscillate the same way), a layer of liquid crystals (with transistors at each pixel that can twist the crystals to change the polarization of light as a function of the data to be displayed) and a layer of color filters.

Today, several variants of LCDs are in use, most of which use white LED backlights. Some recent displays use Quantum-dot LED based LCDs (called QLEDs) which employ blue (instead of white) LEDs and nanocrystals of various sizes to convert light into different colors by altering its wavelength. Thin-film-transistor LCDs (TFT-LCDs) are widely used in large displays, such as HDTVs, computer monitors, and smart home appliances with displays. Having been around a long time, these displays have reached production maturity and thus cost less than other options, but have very poor energy efficiency and viewing angles. In-plane switching LCDs (IPS-LCDs) use a different crystal array orientation and electrical excitation approach for crystals to improve viewing angles and lower power over

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MDAT.2020.2976669, IEEE Design and Test

TFT-LCDs; this type of display has been used in many recent smartphones such as LG G7, Nokia 7 Plus, and Apple iPhone XR.

For LCD displays, energy reduction techniques involve backlight reduction (as it is the most significant factor in LCD energy consumption), as well as dynamic tone mapping, and frame buffer and refresh rate management. Early work in [50] explored reducing backlight energy for video playback. A middleware-based strategy was designed to adaptively reduce backlight levels, while compensating the luminance in video frames, such that output quality for the user was maintained. The strategy was shown to save 100 mW to 625 mW, depending on the type of video and the initial backlight setting, on a handheld Compaq iPAQ device. A histogram equalization approach was proposed in [51] for pixel-level transformation through dynamic tone mapping (mapping high dynamic range of luminance in the real world to a limited range on a display) based on distortion balancing and power management. The technique was extended to video streaming applications with a human visual system aware [52] and algorithmic [53] scaling of backlight dynamically. In contrast to global backlight dimming approaches that control the luminance of all pixels on the screen at the same time, e.g., [54], a local trimming approach was proposed in [55] that divides the entire screen into several blocks and pixels in each block are adjusted separately. Luminance reduction for regions of non-interest on the screen (from a human perception perspective) was proposed in [56] with a neuromorphic saliency model. An approach to exploit change blindness in humans to reduce backlight levels gradually during usage was exploited in [57]. As LCD power consumption is also affected by the frame buffer refresh operations, [58] proposed a frame buffer compression model to minimize energy. Reduction of redundant frames for further energy savings was explored in [59].

OLED Displays

Unlike LCDs, OLED displays do not need a backlight. Instead, each pixel (or subpixel of red, green, or blue) lights itself up as a voltage is applied to a complex molecule called an organic light emitting diode. OLEDs have greater contrast ratios than LCDs and can be flexible (as they do not need a backlight layer and can be thinner), allowing for their use in bendable and foldable phones and devices. Brightness at each pixel is controlled by the value of voltage applied, but comparisons have shown that OLED screens are usually less brighter than LED LCD screens. OLEDs also have lower power consumption than LCDs when displaying mostly dark content, but when displaying mostly light material (such as the common dark text on a light or white background), their power consumption can be much higher than that of an LCD/backlight combination. This explains the interest in "dark" display modes making their way into Android and iOS recently.

There are two types of OLED displays: passive matrix (PMOLEDs) and active matrix (AMOLEDs). PMOLED displays uses a simple control scheme in which each row (line of pixels) is controlled sequentially, one at a time, whereas AMOLED control uses a thin-film transistor (TFT) backplane to directly access and switch each individual pixel on or off, allowing for higher resolution and larger display sizes. PMOLEDs consume more power than AMOLEDs, due to the power needed for their external circuitry. AMOLEDs also provide faster refresh rates than PMOLEDs. Due to these benefits, most mid- to high-end recent smartphones use AMOLED displays, e.g., Samsung Galaxy S10/S10+, Apple iPhone XS, and Google Pixel 4/4XL.

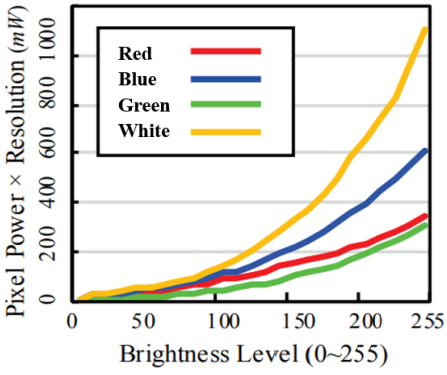

Fig 2. Power consumption for OLED display on Samsung Galaxy S5 smartphone [60].

As an OLED display is self-emitting, power consumption is decided by the pixel color component and brightness, as shown in Fig. 2. Thus various pixel dimming and color transformation algorithms have been proposed to enhance the energy efficiency of OLED displays. The goal is to reduce the power consumption without changing the visual quality perceived by human eyes.

Several efforts have proposed techniques to reduce the power consumption of OLED displays for graphical user interfaces (GUIs). For example, a color transformation-based method is proposed in [61] that transforms the colors of a GUI into new color combinations that require lower power consumption for OLED displays. A smart interaction GUI is designed in [62] to dim the pixels covered by finger shadows. In [63], an approach was proposed to turn OFF selective subpixels to display a GUI with a lower resolution [63]. A shiftmask is proposed in [64] to dim the window while a user is scrolling the page on a browser. However, GUIs are just a small part of the displayed content, and thus such approaches need to be complemented with other methods.

Other methods focus on power savings when viewing images on OLED displays. In [65], a method is proposed that involves image overexposure correction by modifying pixel luminance and chrominance characteristics, in a manner that reduces power consumption on OLED displays. Histogram equalization is used in [66] to increase the contrast of the displayed image, to obtain more scope for pixel dimming. The approach in [67], utilizes Itti's visual-attention model to dim pixels in regions with lower saliency (i.e., regions that are less noticeable in an image by a human). A dynamic voltage scaling (DVS) approach was proposed in [68], to control the supply voltage and reduce power consumption at the circuit level for viewing images on OLED displays. But such DVS-based methods require

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MDAT.2020.2976669, IEEE Design and Test

custom display driver circuitry and cannot be applied to off-the-shelf OLED displays.

A few methods focus on saving power on OLED displays for video content. An approach for adaptive OLED power management was proposed in [69] to control the brightness of areas that are not of interest to the user playing a game. In [60], a framework for low-power OLED-friendly video recording and playback was proposed, with algorithms for pixel dimming, and color range and tone mapping. In [70] the structural similarity index (SSIM) is adopted to guide the fine-grained DVS and color compensation for video streaming. An approach to reduce the brightness of video frames while limiting visual impact via a color blending method was proposed in [71].

4 Wireless Radio Optimizations

Mobile and IoT devices today have several wireless radio components, including those for Wi-Fi, cellular (4G LTE/5G), GPS, Bluetooth, NFC, etc. These components consume significant power and energy, especially during streaming audio/video, multiplayer gaming, and navigation. Thus these components have been the target of much focus and research to minimize their energy footprint.

Location Sensing

Some efforts, e.g., [72], [73], [74] have focused on energy-efficient location-sensing schemes aiming to reduce high energy consumption caused by location interfaces (e.g. Wi-Fi, GPS) by deciding when to enable/disable location interfaces or modify location acquisition frequency. A Variable Rate Logging (VRL) mechanism is proposed in [75] that disables location logging or reduces the GPS logging rate by detecting if the user is standing still or indoors. An adaptive location-sensing framework is proposed in [76] that involves substitution, suppression, piggybacking, and adaptation of an application's location-sensing requests to conserve energy. In [77], the LearnLoc framework was proposed to trade-off energy with localization accuracy, for indoor localization with smartphones. LearnLoc utilized k-nearest neighbor and neural network based machine learning models to predict indoor locations using Wi-Fi signal strength and inertial sensor data, with the ability to vary Wi-Fi sampling rate, where a lower sampling rate was shown to save energy but at the cost of reduced localization accuracy, and vice versa.

Interface Selection

The selection of the wireless interface for data communication (e.g., Wi-Fi vs 4G) has a significant impact on energy consumption. In [78], [79], techniques were proposed to select the most energy-efficient data interface for wireless communication. Some studies compare specific wireless interfaces (e.g., Zigbee and Bluetooth LE [80]) to determine the most appropriate interface to use under different conditions. The Bluesaver framework was proposed in [81], which enabled low latency and low energy wireless communication on mobile devices by maintaining a Bluetooth and Wi-Fi connection simultaneously and switching between them, at the MAC layer. In [82], an approach was proposed to reduce energy-hungry interactions between smartphones and smartwatches. A notification manager was designed to automatically defer "phone-preferable" notifications that require a user to take further actions (such as checking detailed content and replying to a message) and piggyback them on "watch-preferable" notifications that can be handled on a watch, without further interaction with the phone.

In [83] a comprehensive solution was proposed for both data interface selection and location interface optimization, to reduce energy consumption on mobile devices. The proposed middleware framework explores various machine learning techniques towards learning and then predicting the data and location usage requirements for a mobile user, based on their spatiotemporal and device contexts. The context information refers to the user's device usage as a function of the user's location, time of day or week, and specific application being used. Different machine learning algorithms were explored, to learn and predict these contexts, including linear discriminant analysis, linear logistic regression, k-nearest neighbor, non-linear logistic regression with neural networks, and support vector machines. Together with a device power model that was also developed, the context predictions allowed a user- and context-specific selection of optimal strategies for data interface and location interface selection and optimization. It was shown that up to 85% energy savings could be achieved for minimally active users

Component Enhancements

Individual wireless interfaces can also be optimized for low power operation. For instance, the 802.11 power saving mode (PSM) [84] allows putting a wireless radio into a low-power sleep mode whenever it encounters inactive periods. Many variants of PSM have been proposed to dynamically adjust sleep mode periods based on traffic patterns [85], or extend the sleep mode to intervals between packets for devices with light workloads [86]. In [87], a solution is proposed to allow energy-constrained devices to scale down their Wi-Fi sampling rates (i.e., Wi-Fi down-clocking [88]) regardless of channel conditions, to improve energy-efficiency. This is accomplished by making Wi-Fi access points transmit packets with incrementally increasing redundancy until successful reception occurs at a device with a low sampling rate. In [89], passive Wi-Fi was introduced, which allowed the generation of 802.11b transmissions using backscatter [90] communication. The proposed system has two main components: a plugged-in device and passive Wi-Fi devices. The former contains power consuming RF components including a frequency synthesizer and power amplifier, and emits a single tone RF carrier. It also performs carrier sense on behalf of the passive Wi-Fi device and helps coordinate medium access control across multiple passive Wi-Fi devices. The passive Wi-Fi device backscatters the tone emitted by the plugged-in device to create 802.11b transmissions that can be decoded on any device that has a Wi-Fi chipset. It was shown that the proposed system consumed 4–5 orders of magnitude lower power than conventional Wi-Fi, Bluetooth LTE and Zigbee chipsets.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MDAT.2020.2976669, IEEE Design and Test

5 Memory and Storage Optimizations

Main Memory

Mainstream computers today use DDRx DRAM as main memory. Due to the high power footprint of DDRx, mobile and IoT devices use low-power DDR (LPDDR) main memory. LPDDR4, which is widely used in mid- to high-end smartphones today, is optimized for low-power operation, as well as rapid transitioning between various power saving voltage/frequency states. It also supports a feature called partial array self-refresh (PASR) which enables the DRAM to retain state in only part of the memory, thus reducing self-refresh power. There have also been efforts to further reduce main memory energy consumption. For example, a new wide-IO 3D DRAM architecture was proposed in [91] for energy-efficient memory accesses on resource limited devices. In [92], a technique to reduce the refresh power in mobile main memory was proposed. It was shown that mobile devices are idle most of the time, therefore reducing refresh power in idle mode for main memory is essential to reduce energy. The frequency of refresh operations in memory can be reduced significantly by using strong multi-bit error correction codes (ECC), but this incurs a high performance overhead. To obtain both low refresh power in idle periods and high performance in active periods, a Morphable ECC (MECC) approach was utilized. During idle periods, MECC keeps the memory protected with 6-bit ECC and employs a refresh period of 1 second, instead of the typical refresh period of 64ms. During active operation, MECC reduces the refresh interval to 64ms, and converts memory from ECC-6 to weaker single-bit ECC, thus avoiding the high latency of ECC-6. The transition to idle mode for a 1 GB memory with 16 million lines is shown to take 640 million cycles or 400ms to perform the ECC-6 conversion for all lines. As this is a high overhead, an approach is proposed which tracks only the memory that was accessed in the active state and performs an ECC upgrade for only those (accessed) regions of memory. The structure to track the accessed regions of memory has 1K entries (128B), and reduces the transition time to idle mode from 400ms to 50ms. Simulation results indicated that whereas on average strong ECC causes a slowdown of 10% (as high as 21%), with the proposed MECC approach, the average slowdown is reduced to 1.2%. MECC was also shown to reduce refresh power in idle periods by 16X and idle power by 2X.

Secondary Storage

While secondary storage is not a significant power consumer in mobile and IoT devices, its long access latencies can result in a notable energy consumption. Studies performed on the Samsung Galaxy Nexus S smartphone in [93] indicated that for IO intensive workloads with predominantly random accesses, more than 30% of the energy can be consumed in the storage system (to access the smartphone's internal eMMC flash storage). While one could argue that the IO intensive workloads are not representative of common applications used in IoT and mobile devices, the study does highlight the contribution of memory and storage, which can matter for at least some applications. MobiFS [94] proposes trading durability for improved memory energy efficiency in smartphones. This is accomplished by reducing the amount of data flushed to flash storage, and relaxing the timing of flush operations, at the risk of data loss due to system failures or power loss, which would be rare in most mobile and IoT devices. Flashlogger [95] proposed using amnesic compression techniques to lower energy costs for storage in sensor systems. A fast storage system based on battery-backed RAM to increase the performance and energy-efficiency of wearables was proposed in [96]. Smartwatch storage energy was explored in [97] where it was shown that the amount of data written daily is 10× as large as the amount of data read daily, and the amount of data written to the flash storage each day is approximately as large as the free space in the storage device. To minimize the energy consumption associated with the IO activities in the smartwatch, new file system management algorithms were proposed that reduced flush operations to flash. This was shown to reduce overall smartwatch energy by 3% and IO energy by 60%.

6 Software Optimizations

More than 5 million apps for smart devices have been developed across the Apple App store and Google Play store as of the end of 2019 [98]. Since each app consists of different resource requirements, constructing a unified resource management policy for all of them is challenging. Therefore, a variety of software and operating system (OS)-level energy management techniques have been proposed in literature [99], [100].

Runtime software profiling can identify the most power and energy hungry resources that should be targeted by dynamic management approaches. To this end, the Powerscope tool profiles energy usage of active applications by mapping energy consumption to program structure [101]. It combines hardware instrumentation to measure current levels with kernel software support to perform statistical sampling of system activity. Pathak et al. [102] proposed power modeling based on system call tracing using both utilization and non-utilization-based power behavior. The Appscope tool monitors applications' hardware usage at the kernel level to further improve the accuracy and observability [103]. More recently, a purely software-based energy profiling tool is proposed for Android apps [104]. The authors demonstrate the practicality and accuracy of software implementation against hardware measurements. A detailed survey on software energy consumption and the potential research directions is presented in [105].

Early work on mapping software applications to heterogeneous PEs focused on static [106] and runtime [107] techniques, assuming multiple voltage-frequency levels. Similarly, Khdr et al. [108] constructed an approach to assign applications to tiles in multi-core architectures depending on the degree of parallelism and available voltage-frequency levels of each tile. While these approaches are well suited for mapping applications onto a given resource, there is still need for techniques which help to improve performance at the operating system level.

Numerous researchers have addressed the need of thermal and power management techniques at the OS level. A

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MDAT.2020.2976669, IEEE Design and Test

hybrid of hardware and software techniques, HybDTM, was proposed to lower system temperature [109]. This technique utilized regression-based thermal models to estimate the system temperature at runtime. A scheduling technique at the OS level was proposed to reduce the on-chip temperature [110]. The authors first presented a systematic study to show that the rise and fall-time of the temperature are more than 10x higher than OS scheduler ticks. Within this time-constant, tasks which are responsible to increase system-temperature (hot tasks) are migrated to an idle core. If the system utilization is high, then each core is assigned hot and cold tasks. While this technique reduces only the peak temperature, a recent scheduling technique reduces both peak and average temperature of the system [111]. Recently, another OS-level algorithm computes the maximum power budget by predicting temperature over a time horizon [112]. Then, this power budget is used to turn off cores and throttle core frequencies to avoid temperature violations. Similarly, a software level thermal management technique for DRAM is proposed in [113].

Apart from thermal management, different dynamic management techniques at OS-level are also discussed in the literature. Snowdon et al. propose a platform that can execute different power management policies [114]. This platform showed significant energy savings with minimal performance loss when integrated with Linux kernel. The authors in [115] proposed modifications to the runtime power management framework in the Linux OS. In this technique, all power management drivers are replaced with a centralized agent. Zhang et. al [116] implemented a hybrid of software and hardware-oriented power management techniques on Linux/x86 platforms. This technique provides competitive performance with respect to Intel's commercial hardware platforms. A more detailed study on software-oriented energy management techniques for heterogeneous mobile systems is presented in [117].

7 Cloud Offloading and Distribution of Computation across IoT Networks

Cloud offloading became ubiquitous with the adoption of smartphones and tablets. e.g., most of the speech recognition systems today offload some or a majority of the tasks to cloud servers and accelerators. Others train models in the cloud and use inference on the mobile device. More generally in IoT and mobile devices, machine learning (ML) and deep-learning (DL) is one of the major applications for studies of optimal offloading and distribution of the computation and the communication (thanks to the popularity of the domain and ease of structural decomposition of the inference computation).

With the explosion of mobile applications and the support of cloud computing for a variety of services for mobile users, mobile cloud computing (MCC) has been introduced as an integration of cloud computing into the mobile environment. MCC brings new types of services to take advantage of cloud computing and supports static and dynamic offloading decisions [118], [119]. As an example, [120] proposed a framework for offloading computation from apps to the cloud in a dynamic and opportunistic manner, while considering factors such as the Wi-Fi or cellular network channel conditions and the compute/communication characteristics of the application. A reinforcement learning based middleware approach was proposed for decision making. Results of using this framework across multiple apps from the Android app store showed a reduction of up to 30% in energy consumption and also improvements in the app response time.

An offloading strategy to optimize the performance of IoT deep learning applications with edge computing was proposed in [121]. Parts of the layers of the deep learning network used for inference (in this case for video data recognition) are scheduled for execution on edge servers, while the other layers are scheduled for cloud computation. This idea can be used for both a static decision and on-line dynamic scheduling that changes the distribution of layers for offloading depending on the complexity and number of images. In [122] the authors push ML inference computation out of the cloud onto a hierarchy of IoT devices. They developed a refactoring algorithm for ML inference computation to factor it over a network of devices without significantly reducing prediction accuracy while also exploring ML model approximations. The approach significantly reduces system energy without reducing accuracy relative to sending all of the data to the cloud. In [123] a cloud offloading scheduler is designed based on a Lyapunov optimization scheme. They derive an online algorithm and prove performance bounds for average power consumption and also queue occupancy (which is indicative of delay).

The introduction of 5G wireless communication has led to a further explosion in the amount of computation and communication in IoT networks [124]. Multi-edge computing (MEC) technology [125], [126] has been introduced to offer cloud-computing capabilities within the radio access networks and help to satisfy 5G latency requirements. MEC assumes that distributed nodes can be placed adjacent to end devices (device edge compute nodes) and also closer to the cloud (cloud edge compute nodes on the networks at the periphery of the mobile network and data centers). Typically, remote task offloading incurs large over-the-air transmission and computing delays. The pressure to reduce latency in computing and decision making is being driven by 5G deployment especially in the ultra-reliable low latency communication (URLLC) domain. The URLLC domain has latency requirement of 1ms and requirement for reliability of transmitting within 1ms of 99.99999%. The latency and reliability requirements are critical for vehicle-to-everything (V2X) use cases [124] as well as some industrial IoT applications. These requirements lead to the need for more optimal distribution of computation and communication across multiple nodes of the IoT network: end nodes and devices, a device edge, a cloud edge and the cloud data centers. Some recent work has begun to explore offloading for energy-efficient mobile edge computing over 5G networks [127].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MDAT.2020.2976669, IEEE Design and Test

8 Battery-Aware Design

Battery lifetime is a paramount metric in battery-powered mobile and IoT devices. Longer battery lifetime brings numerous benefits, such as better user experience, reduced maintenance costs in industrial IoTs, and smoother operation of the IoT networks. Modern devices are equipped with batteries that have bounded energy capacity, usually expressed in Ampere-hours. Enhancing battery lifetime can be attained via tailoring power management algorithms to the characteristics of the battery technology as well as improved energy efficiency of the system.

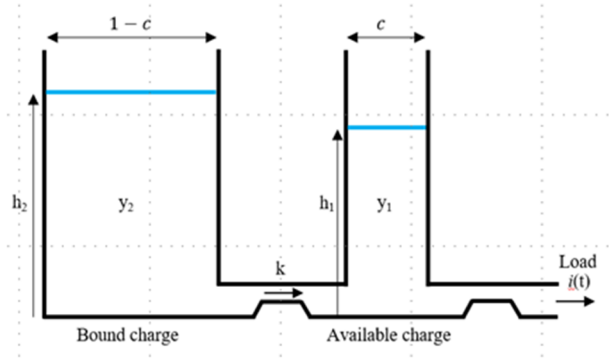

Fig 3. Two-well model of the Kinetic battery model. The value of c corresponds to the fraction of total capacity is placed in the available charge well (y1) and k is the rate constant [128].

Battery Modeling

The lifetime of the battery depends not only on the rate of which the energy is consumed but also on the dynamics of the load current. In general, higher load current leads to a drop in residual capacity of the battery, whereas idle periods help in partial capacity recovery [128]. In consequence, the energy and voltage delivered by the battery depend heavily on the usage pattern. Detailed and abstract battery models have been developed over the years which describe the relationship between power consumption and the state of the battery. In [129], [130] detailed electrochemical models are proposed which represent the battery using a set of coupled non-linear differential equations that require numerical solutions. Although detailed models are highly accurate, they are not suitable for runtime optimizations due to their heavy computational overhead. Several high-level models have been developed that provide a good trade-off between runtime computational overhead and accuracy. The widely adopted analytical models are the kinetic battery model (KBM) [131] and the diffusion-based models [132] which approximate the non-linearity of the battery using a set of two differential equations that can be solved analytically. The KBM is an intuitive model which distributes the battery change into two wells, the available charge well and the bound charge well as shown in Figure 3. The available charge well delivers the load current, while the bound charge well feeds electrons only to the avail-able charge well [128], [131].

Battery-aware Resource Management

Several dynamic processing task scheduling techniques have been proposed that aim to elongate battery lifetime while meeting desired performance targets. One class of techniques have focused on edge mobile devices. A dynamic scheduling algorithm was proposed in [133] for devices running periodic task graphs with real time deadlines constraints. This algorithm orchestrates the Dynamic Voltage and Frequency Scaling (DVFS) assignments of the processor and task scheduling while maximizing battery lifetime. As part of this work, the authors proposed a set of battery aware scheduling guidelines derived from the KBM battery model, for example, scheduling the voltage and clock speed assignments locally in a non-increasing order. In [134], a DVFS algorithm was introduced to optimize for the total battery current of embedded systems equipped with a multiprocessor that runs concurrent tasks. The work in [135] focused on delivering battery lifetime guarantees for a selected set of applications and best effort for the remaining ones, targeting smart mobile devices. This technique periodically profiles the battery usage pattern of the running applications and determines the battery budget required for the priority applications to meet their performance deadlines. In [136], a CPU-GPU control algorithm is presented that enhances the energy efficiency of mobile devices to improve battery lifetime. This algorithm controls the DVFS of the CPU and GPU subsystem in a coordinated manner to deliver a desired video frame per second (FPS) performance while minimizing energy consumption. On the network side, the authors in [137], [138] introduced techniques to maximize battery lifetime at the network level which utilize battery models to guide the scheduling decisions.

9 User-Aware Optimizations

The usage profile of mobile smart devices has evolved over the years and today these devices have become an essential part of our daily life. This transformation has been driven by continued growth in the sophistication and functionality of applications. It has resulted in an ever closer interaction between users and their smart devices. User experience has emerged as a fundamental metric in the realm of smart devices. Unsurprisingly, user-experience is a multi-dimensional metric and usually user dependent. This metric cannot be ignored when considering optimizations for energy-efficiency in mobile devices.

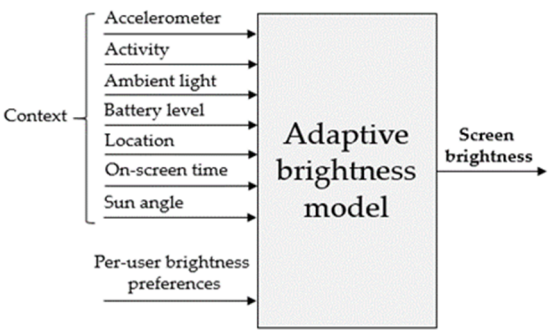

Fig 4. On-line learning control of display brightness [139].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MDAT.2020.2976669, IEEE Design and Test

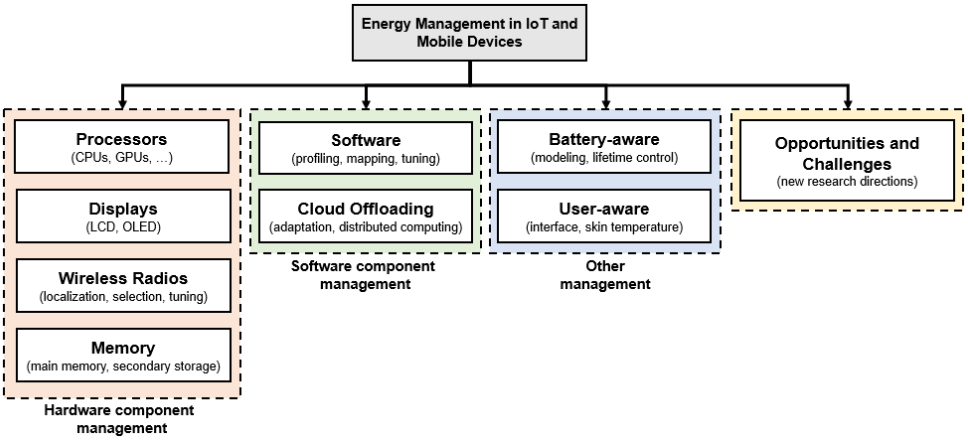

Fig 5. Summary of survey organization.

The display is the main user interface for mobile systems. Both brightness and content of the display impact user satisfaction. The authors in [139] introduced an online learning algorithm that dynamically controls the display brightness to meet individual users' preferences. Figure 4, depicts a high-level description of this control. It takes runtime contextual data (e.g. ambient light, battery level, location) as part of the input in addition to occasional per-user brightness preference feedback to improve prediction accuracy of brightness preference of the user. The work in [140] presented a technique to minimize power by reducing the frame rate of the display to the tolerated level by the user. It leverages the intrinsic variations in people's tolerance to frame rates and introduced an adaptive algorithm that dynamically predicts the tolerated frame rate of individual users. In [141] the authors adopted a similar concept for the CPU subsystem where the goal was to minimize the CPU processing rate while ensuring user satisfaction.

Skin temperature is another important metric for user experience in the realm of smart devices. The significance of this metric is driven from the non-trivial amount of heat that can be generated from these devices during normal usage when the users are often in contact with the device surface. Consequently, high skin temperature may lead to discomfort or even cause harm to the users' skin. The skin temperature can be maintained within the desired threshold using thermal management algorithms which usually implement close loop feedback control. Such control requires feedback from skin temperature where direct measurements are usually not feasible in practice due to practical limitations. The alternative solution is to estimate the temperature using abstract thermal models. In [142], [143] the authors proposed machine learning based models to estimate the skin temperature of mobile devices that take inputs from internal sensors and hardware counters of the device. The work in [142] also proposed a user-aware skin temperature thermal management strategy. It employed user dependent skin temperature thresholds to improve the user experience, based on the observation that the sensitivity of users to skin temperature varies across users.

10 Opportunities and Challenges

The preceding sections have discussed the state-of-the-art and trends with various modalities of energy optimization for mobile and IoT devices. Figure 5 summarizes the scope of the survey. In this section, we summarize some of the emerging directions for energy minimization in such devices and opportunities for new research:

Processing and Software Optimizations: Two critical components of all energy management techniques are (1) the ability to accurately observe the power consumption of all major resources and (2) control their power states independently. The former requires power meters, implemented either in software or using current sensor. While dedicated resources increase the observability, they also incur significant implementation overhead. Similarly, individual voltage-frequency islands improve controllability, again with larger overhead. Hence, energy efficiency should be maximized with limited observability and controllability. This requires new approaches for co-optimizations in hardware architectures (e.g., multiple core clusters with different power-performance trade-offs), software (e.g., matching the power states of the clusters with application requirements), and firmware support to interface the hardware knobs with the OS.

Displays: We are gradually moving towards devices with flexible displays that can be bent and folded, for which variants of the OLED technology are a good match. However, OLED displays are prone to burn-ins. Mini LED displays and eventually Micro LED displays that utilize miniaturized LED arrays that require an ultra-thin backlight layer or no backlight at all, respectively, could lead to very low power and flexible displays that do not suffer from burn-in issues, in mobile and IoT devices. E-paper displays (also sometimes called e-ink displays) provide another low energy option for many IoT and mobile applications. These displays use millions of tiny capsules that contain black and white ink particles. A charge applied to the top and bottom of a capsule arranges the ink particles to form an image or text. No backlight is needed as the displayed content is visible with reflected light, allowing e-paper dis-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MDAT.2020.2976669, IEEE Design and Test

plays to be much thinner than TFT LCD displays. However, it is possible to include a backlight as well as a touchscreen with these displays, e.g., as done in Amazon's Kindle Paperwhite and the Barnes and Noble Nook e-readers. Color e-paper displays with high resolution are emerging, providing another ultra-low power display solution. These new technologies will benefit from new approaches to more aggressively reduce their power overheads in mobile and IoT devices.

Wireless radios: While long-range Wi-Fi and shorter-range Bluetooth LE and Zigbee standards will continue to be popular for low-cost wireless communication with mobile and IoT devices, new standards are emerging. Low Power Wide Area Network (LPWAN) standards such as LoRaWan will allow IoT devices to be seamless connected over long distances (several kilometers) for low-bit rate communication. The rollout of 5G will also enable ultra-high bandwidth communication over long distances, but new techniques will be essential to minimize power consumption for participating devices.

Storage: The maturation of non-volatile memory technologies based on spin-transfer torque effects, and memristive effects will allow for opportunities to collapse the traditional deep memory hierarchy into a shallower one, with non-volatile memory elements being closely integrated within processor dies and on die-stacks. New research is needed to optimize such memories for energy-efficient operation with mobile and IoT workloads.

Energy harvesting-aware design: Battery charging and replacement remain among the leading factors that deteriorate user experience. This challenge can be addressed by pioneering research in two directions that complement the energy management techniques surveyed in this paper. First, ambient energy sources, such as, light, body heat, radio frequency and motion, can be exploited to replace or complement the battery energy. Second, runtime energy management techniques can match the consumed energy with the available energy to provide uninterrupted operation while optimizing the user experience.

Cloud offloading and distribution of computation across IoT networks: End-to-end IoT systems contain multiple levels of hierarchy. The pressure to reduce latency in computing and decision making driven by the 5G deployment especially in the ultra-reliable low latency communication pushes the need for smarter distribution of computation and communication both at the time of system design and at run-time. Research on optimal offloading and distribution of computation and communication from performance, quality of service guarantees, energy, reliability, and safety is required. To enable the research on end-to-end exploration and optimization of such large scale systems it is critical to develop simulation infrastructures for hierarchical IoT systems with sufficiently accurate models for the above metrics. [144] is one of the recent attempts for developing such a framework.

ACKNOWLEDGMENTS

This research was supported in part by the National Science Foundation (NSF) under grant numbers ECCS-1646562 and CCF-1813370, Semiconductor Research Corporation (SRC) task 2721.001, and Strategic CAD Labs, Intel Corporation.

REFERENCES

[1] X. Chen, et al. "Smartphone energy drain in the wild: Analysis and implications." *ACM SIGMETRICS Performance Evaluation Review 43.1 (2015): 151-164.*

[2] ARM Cortex CPUs, [Online]: https://www.arm.com/products/silicon-ip-cpu

[3] Adreno™ Graphics Processing Units, [Online]: https://developer.qualcomm.com/software/adreno-gpu-sdk/gpu

[4] JEDEC Updates Standard for Low Power Memory Devices: LPDDR5 [Online]: https://www.jedec.org/standards-documents/docs/jesd209-5

[5] Wide Range Vcc Flash, [Online]: https://www.macronix.com/en-us/products/NOR-Flash/Pages/Ultra-Low-Power-Flash.aspx

[6] Transmission of IPv6 Packets Over IEEE 802.15.4 Networks. [Online]: https://tools.ietf.org/html/rfc4944

[7] LoRa Alliance, [Online]: https://lora-alliance.org/

[8] RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. [Online]: https://tools.ietf.org/html/rfc6550

[9] Bluetooth Low Energy (BLE) [Online]: https://www.bluetooth.com/bluetooth-technology/radio-versions/

[10] 3GPP: Release 12. [Online]: http://www.3gpp.org/specifications/releases/68-release-12

[11] Contiki: The Open Source Operating System for the Internet of Things. [Online]: http://www.contiki-os.org/

[12] P. A. Levis, "TinyOS: An open operating system for wireless sensor networks," *in Proc. 7th Int. Conf. Mobile Data Manage. (MDM), May 2006, p. 63.*

[13] FreeRTOS: Quality RTOS&Embedded Software. [Online]: http://www.freertos.org/

[14] Zephyr real-time RTOS, [Online]: https://github.com/zephyrproject-rtos/zephyr

[15] Android OS, [Online]: https://www.android.com/

[16] Hardkernel. ODROID-XU3: https://wiki.odroid.com/old_product/odroid-xu3/odroid-xu3

[17] D. Foley, et al. "A low-power integrated x86–64 and graphics processor for mobile computing devices." *IEEE Journal of Solid-State Circuits 47.1 (2011): 220-231.*

[18] U. Gupta, et al. "Dynamic power budgeting for mobile systems running graphics workloads," *IEEE Transactions on Multi-Scale Computing Systems, 4(1), 2017, pp. 30-40.*

[19] D. Kadjo, et al. "Towards platform level power management in mobile systems." *IEEE International System-on-Chip Conference, 2014, pp. 146-151.*

[20] Qualcomm® Snapdragon™ 855+ Mobile Platform. [Online]: https://www.qualcomm.com/products/

snapdragon-855-plus-mobile-platform

[21] V. Pallipadi, and A. Starikovskiy. "The ondemand governor." *in Proc. of the Linux Symposium. Vol. 2. No. 00216. 2006.*

[22] U.Y. Ogras, R. Marculescu, D. Marculescu, E.G. Jung, "Design and management of voltage-frequency island partitioned networks-on-chip," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems, 17(3), 330-341, 2009.*

[23] N. Vallina-Rodriguez and J. Crowcroft. "Energy management techniques in modern mobile handsets," *IEEE Communications Surveys & Tutorials 15.1 pp. 179-198, 2012.*

[24] R. David, et al. "Dynamic power management of voltage-frequency island partitioned networks-on-chip using intel's single-chip cloud computer," *in Proc. of International Symposium, pp. 257-258, 2011.*

[25] A. Pathania, et al. "Integrated CPU-GPU power management for 3D mobile games." *in Proc. 51st ACM/EDAC/IEEE Design Automation Conference (DAC). IEEE, 2014.*

[26] B. Donyanavard, et al. "Sparta: Runtime task allocation for energy efficient heterogeneous manycores." *In Proc. International Conference on Hardware/Software Codesign and System Synthesis (CODES+ ISSS), 2016.*

[27] A.K. Singh, et al. "Energy-efficient run-time mapping and thread partitioning of concurrent opencl applications on cpu-gpu mpsocs." *ACM Transactions on Embedded Computing Systems (TECS) 16.5s (2017): 147.*

[28] Q. Qiu and M. Pedram. "Dynamic power management based on continuous-time Markov decision processes." *Proceedings 1999 Design Automation Conference (Cat. No. 99CH36361). IEEE, 1999.*

[29] A. Aalsaud, et al. "Power--aware performance adaptation of concurrent applications in heterogeneous manycore systems." *Proceedings of the 2016 International Symposium on Low Power Electronics and Design. ACM, 2016.*

[30] R. Cochran, et al. "Pack & Cap: adaptive DVFS and thread packing under power caps." *in Proc. 44th Annual International Symposium on Microarchitecture (MICRO), 2011.*

[31] G. Dhiman, "System-level power management using online learning." *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 28.5 (2009): 676-689.*

[32] P. Bogdan and R. Marculescu. "Workload characterization and its impact on multicore platform design," *in Proc. International Conference on Hardware/Software Codesign and System Synthesis. IEEE, 2010.*

[33] U. Gupta, et al. "Dypo: Dynamic pareto-optimal configuration selection for heterogeneous mpsocs." *ACM Transactions on Embedded Computing Systems (TECS) 16.5s (2017): 123.*

[34] D. Silver, et al. "Mastering the game of go without human knowledge." *Nature 550.7676 (2017): 354.*

[35] Z. Chen and D. Marculescu. "Distributed reinforcement learning for power limited many-core system performance optimization." *in Proc. Design, Automation & Test in Europe Conference & Exhibition. EDA Consortium, 2015.*

[36] F.M.M. Ul Islam and M. Lin. "Hybrid DVFS scheduling for real-time systems based on reinforcement learning." *IEEE Systems Journal 11.2 (2015): 931-940.*

[37] Q. Zhang, et al. "A double deep Q-learning model for energy-efficient edge scheduling." *IEEE Transactions on Services Computing, 2018.*

[38] U. Gupta, et al. "A deep q-learning approach for dynamic management of heterogeneous processors." *IEEE Computer Architecture Letters 18.1: 14-17, 2019.*

[39] S. Schaal, "Is imitation learning the route to humanoid robots?." *Trends in cognitive sciences 3.6 (1999): 233-242.*

[40] S. Ross, G. Gordon, and D. Bagnell. "A reduction of imitation learning and structured prediction to no-regret online learning." *in Proc. international conference on artificial intelligence and statistics. 2011.*

[41] R.G. Kim, et al. "Imitation learning for dynamic VFI control in large-scale manycore systems." *IEEE Transactions on Very Large Scale Integration (VLSI) Systems 25.9 (2017): 2458-2471.*

[42] S.K. Mandal, et al. "Dynamic resource management of heterogeneous mobile platforms via imitation learning." *IEEE Transactions on Very Large Scale Integration (VLSI) Systems 27.12 (2019): 2842-2854.*

[43] D. Kadjo, et al. "A control-theoretic approach for energy efficient CPU-GPU subsystem in mobile platforms." *in Proc. 52nd Annual Design Automation Conference. ACM, 2015.*

[44] B. Dietrich and S. Chakraborty. "Lightweight graphics instrumentation for game state-specific power management in Android." *Multimedia Systems 20.5 (2014): 563-578.*

[45] U. Gupta, et al. "An Online Learning Methodology for Performance Modeling of Graphics Processors." *IEEE Transactions on Computers 67.12 (2018): 1677-1691.*

[46] J-Y Won, et al. "Up by their bootstraps: Online learning in artificial neural networks for CMP uncore power management." *in Proc. IEEE 20th International Symposium on High Performance Computer Architecture (HPCA). IEEE, 2014.*

[47] X. Chen, et al. "In-network monitoring and control policy for DVFS of CMP networks-on-chip and last level caches," *ACM Transactions on Design Automation of Electronic Systems (TODAES) 18, no. 4, 47, 2013.*

[48] S.M. Tam, et al. "SkyLake-SP: A 14nm 28-Core xeon® processor." *in Proc. IEEE International Solid-State Circuits Conference-(ISSCC). IEEE, 2018.*

[49] A. Miele, et al. "On-chip dynamic resource management." *Foundations and Trends in Electronic Design Automation 13.1-2, 2019: 1-14.*

[50] S. Pasricha, M. Luthra, S. Mohapatra, N. Dutt, N. Subramanian, "Dynamic Backlight Adaptation for Low Power Handheld Devices", *IEEE Design and Test (D&T), Special Issue on Embedded Systems for Real Time Embedded Systems, Sep-Oct 2004.*

[51] A. Iranli, H. Fatemi, and M. Pedram, ``HEBS: Histogram equalization for backlight scaling,'' *in Proc. Test Eur. Desing Automat., 2005, pp. 346-351.*

[52] A. Bartolini, M. Ruggiero, and L. Benini, ``HVS-DBS:

human visual system-aware dynamic luminance backlight scaling for video streaming applications," *in Proc. 7th ACM Int. Conf. Embedded Softw., 2009, pp. 21-28.*

[53] P.-C. Hsiu, C.-H. Lin, and C.-K. Hsieh, ``Dynamic backlight scaling optimization for mobile streaming applications,'' *in Proc. 17th IEEE/ACM Int. Symp. Low-Power Electron. Design, Aug. 2011, pp. 309-314.*

[54] S.-J. Kang and Y. H. Kim, "Segmentation-based clipped error control algorithm for global backlight dimming," *J. Display Technol., vol. 10, no. 7, pp. 321–328, Jul. 2014.*

[55] S. Chen and H. Tsai, "A Novel Adaptive Local Dimming Backlight Control Chip Design Based on Gaussian Distribution for Liquid Crystal Displays," *in Journal of Display Technology, vol. 12, no. 12, pp. 1494-1505, Dec. 2016.*

[56] Y. Xiao, K. Irick, V. Narayanan, D. Shin, and N. Chang, "Saliency aware display power management," *in Proc. Conf. Desing, Automat. Test Eur., 2013, pp. 1203-1208.*

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

[58] Y. Jiang, Y. Li, D. Ban, and Y. Xu, "Frame buffer compression without color information loss," *in Proc. IEEE 12th Int. Conf. Comput. Inf. Technol. (CIT), Oct. 2012, pp. 12-17.*

[59] Y. Huang, M. Chen, L. Zhang, S. Xiao, J. Zhao, and Z. Wei, "Intelligent frame refresh for energy-aware display subsystems in mobile devices," *in Proc. Int. Symp. Low Power Electron. Design, 2014, pp. 369-374.*

[60] X. Chen, J. Mao, J. Gao, K. W. Nixon, and Y. Chen, "MORPh: Mobile OLED-friendly recording and playback system for low power video streaming," *in Proc. ACM/EDAC/IEEE Des. Automat. Conf., Jun. 2016, pp. 1–6.*

[61] M. Dong, Y.-S. K. Choi, and L. Zhong, "Power-saving color transformation of mobile graphical user interfaces on OLED-based displays," *in Proc. IEEE Int. Symp. Low Power Electron. and Design, Aug. 2009, pp. 339–342.*

[62] X. Chen, K. W. Nixon, H. Zhou, Y. Liu, and Y. Chen, "FingerShadow: An OLED power optimization based on smartphone touch interactions," *in Proc. 6th Workshop Power-Aware Comput. Syst., Oct. 2014, p. 6.*

[63] P. K. Choubey, A. K Singh, R. B. Bankapur, Vaisakh P. C. SB, and Manoj Prabhu B., "Content aware targeted image manipulation to reduce power consumption in OLED panels," *in Proc. 8th Int. Conf. Contemporary Comput., Aug. 2015, pp. 467–471.*

[64] H.-Y. Lin, P.-C. Hsiu, and T.-W. Kuo, "ShiftMask: Dynamic OLED power shifting based on visual acuity for interactive mobile applications," *in Proc. IEEE/ACM Int. Symp. Low Power Electron. Des., Jul. 2017, pp. 1–6.*

[65] P. Chondro and S.-J. Ruan, "Perceptually hue-oriented power-saving scheme with overexposure corrector for AMOLED displays," *J. Display Technol., vol. 12, no. 8, pp. 791–800, Aug. 2016.*

[66] L. M. Jan, F. C. Cheng, C. H. Cheng, S. J. Ruan, and C. A. Shen, "A power-saving histogram adjustment algorithm for OLED-oriented contrast enhancement," *J. Display Technol., vol. 12, no. 4, pp. 368–375, Oct. 2016.*

[67] C.-H. Lin, C.-K. Kang, and P.-C. Hsiu, "CURA: A framework for quality retaining power saving on mobile OLED displays," *ACM Trans. Embedded Comput. Syst., vol. 15, no. 4, Aug. 2016, Art. no. 76.*

[68] D. Shin, Y. Kim, and M. Pedram, "Dynamic voltage scaling of OLED displays," *in Proc. Design Autom. Conf., Jun. 2011, pp. 53–58.*

[69] T. K. Wee and R. K. Balan, "Adaptive display power management for OLED displays," *ACM SIGCOMM Comput. Commun. Rev., vol. 42, no. 4, pp. 485–490, Oct. 2012.*

[70] S.-J. Kang, "Perceptual quality-aware power reduction technique for organic light emitting diodes," *J. Display Technol., vol. 12, no. 6, pp. 519– 525, Jun. 2016.*

[71] M. Park and M. Song, "Saving power in video playback on OLED displays by acceptable changes to perceived brightness," *J. Display Technol., vol. 12, no. 5, pp. 483–490, May 2016.*

[72] I. Constandache, S. Gaonkar, M. Sayler, R. R. Choudhury, L. Cox, "EnLoc: energy-efficient localization for mobile phones," *in Proc. INFOCOM, pp. 19-25, Jun. 2009.*

[73] K. Lin, A. Kansal, D. Lymberopoulos, F. Zhao, "Energy-accuracy trade-off for continuous mobile device Location," *in Proc. MobiSys, pp. 285-298. Jun. 2010.*

[74] F. B. Abdesslem, A. Phillips, T. Henderson, "Less is more: energy-efficient mobile sensing with SenseLess," *in Proc. MobiHeld, pp. 61-62, Aug. 2009.*

[75] C. Lee, M. Lee, D. Han, "Energy efficient location logging for mobile device," *in Proc. SAINT, pp. 84, Oct. 2010.*

[76] Z. Zhuang, K. Kim, J. P. Singh, "Improving energy efficiency of location sensing on smartphones," *in Proc. MobiSys, pp. 315-330, Jun. 2010.*

[77] S. Pasricha, V. Ugave, Q. Han and C. Anderson, "LearnLoc: A Framework for Smart Indoor Localization with Embedded Mobile Devices," *in Proc. ACM/IEEE International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS), Oct 2015.*

[78] H. Petander, "Energy-aware network selection using traffic estimation," *in Proc. MICNET, pp. 55-60, Sept. 2009.*

[79] M. Ra, J. Paek, A. B. Sharma, R. Govindan, M. H. Krieger, M. J. Neely, "Energy-delay tradeoffs in smartphone applications," *in Proc. MobiSys, pp. 255-270, Jun. 2010.*

[80] J. Lee, M. Dong and Y. Sun, "A preliminary study of low power wireless technologies: ZigBee and Bluetooth Low Energy," *in Proc. IEEE 10th Conference on Industrial Electronics and Applications (ICIEA), Auckland, 2015, pp. 135-139.*

[81] A. Pyles, D. T. Nguyen, X. Qi and G. Zhou, "Bluesaver: A Multi-PHY Approach to Smartphone Energy Savings," *in IEEE Transactions on Wireless Communications, vol. 14, no. 6, pp. 3367-3377, June 2015.*

[82] J. Lee, U. Lee and H. Kim, "PASS: Reducing Redundant Interactions between a Smartphone and a Smartwatch

for Energy Saving," *in IEEE Transactions on Mobile Computing, 2019.*

[83] B. Donohoo, C. Ohlsen, S. Pasricha, C. Anderson, Y. Xiang, "Context-Aware Energy Enhancements for Smart Mobile Devices", *IEEE Transactions on Mobile Computing (TMC), Vol 13, No. 8, pp. 1720-1732, Aug 2014.*

[84] Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Standard, IEEE Std. 802.11, 2007.

[85] R. Krashinsky and H. Balakrishnan, "Minimizing energy for wireless Web access with bounded slowdown," *in Proc. ACM MobiCom, 2002, pp. 119–130.*

[86] F. R. Dogar, P. Steenkiste, and K. Papagiannaki, "Catnap: Exploiting high bandwidth wireless interfaces to save energy for mobile devices," *in Proc. ACM MobiSys, 2010, pp. 107–122.*

[87] W. Wang, Y. Chen, L. Wang and Q. Zhang, "Sampleless Wi-Fi: Bringing Low Power to Wi-Fi Communications," *IEEE/ACM Transactions on Networking, vol. 25, no. 3, pp. 1663-1672, June 2017.*

[88] F. Lu, G. M. Voelker, and A. C. Snoeren, "SloMo: Downclocking WiFi communication," *in Proc. USENIX NSDI, 2013, pp. 255–258.*

[89] B. Kellogg, et al. "Passive wi-fi: Bringing low power to wi-fi transmissions." *in Proc. 13th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 16). 2016.*

[90] B. Kellogg, A. Parks, S. Gollakota, J. R. Smith, and D. Wetherall, "Wi-fi backscatter: Internet connectivity for rf-powered devices", *in Proc. ACM Conference on SIGCOMM, 2014.*

[91] I. Thakkar, S. Pasricha, "3D-WiRED: A Novel Wide I/O DRAM with Energy-Efficient 3D Bank Organization", *IEEE Design and Test (D&T), vol.32, no.4, pp.71-80, Aug. 2015.*

[92] C. Chou, P. Nair and M. K. Qureshi, "Reducing Refresh Power in Mobile Devices with Morphable ECC," *in Proc. Annual IEEE/IFIP International Conference on Dependable Systems and Networks, Rio de Janeiro, 2015, pp. 355-366.*

[93] J. Mohan, et al. "Storage on your smartphone uses more energy than you think." *in Proc. 9th {USENIX} Workshop on Hot Topics in Storage and File Systems (HotStorage 17). 2017.*

[94] J. Ren, et al., "Memory-centric data storage for mobile systems" *in Proc. USENIX Annual Technical Conference, pages 599–611, 2015.*

[95] S. Nath, "Energy efficient sensor data logging with amnesic flash storage", *in Proc. International Conference on Information Processing in Sensor Networks, pages 157–168. IEEE Computer Society, 2009.*

[96] J. Huang, A. Badam, R. Chandra, and E. B Nightingale, "WearDrive: Fast and Energy-Efficient Storage for Wearables", *in Proc. of USENIXATC, 2015.*

[97] J. Kim, et al., "Energy Efficient IO stack Design fo Wearable Device", *in Proc. ACM/SIGAPP Symposium on Applied Computing (SAC'19), April 8–12, 2019.*

[98] Statista. *Mobile App Usage – Statistics & Facts*. [Online]. Available: https://www.statista.com/topics/1002/mobile-app-usage/

[99] N. Vallina-Rodriguez and J. Crowcroft. "ErdOS: achieving energy savings in mobile OS." *in Proc. International Workshop on MobiArch, pp. 37-42. ACM, 2011.*

[100] A. Javed, et al. "Energy Consumption in Mobile Phones." *in Proc.International Journal of Computer Network and Information Security 10, no. 12, 2017: 18.*

[101] J. Flinn and M. Satyanarayanan. "Powerscope: A tool for profiling the energy usage of mobile applications." *in Proc. WMCSA'99. Second IEEE Workshop on Mobile Computing Systems and Applications. IEEE, 1999.*

[102] A. Pathak, et al. "Fine-grained power modeling for smartphones using system call tracing." *in Proc. sixth conference on Computer systems*. ACM, 2011.

[103] C. Yoon, et al. "Appscope: Application energy metering framework for android smartphone using kernel activity monitoring." *in Proc. 2012 {USENIX} Annual Technical Conference ({USENIX}{ATC} 12)*. 2012.

[104] D. Di Nucci, et al. "Software-based energy profiling of android apps: Simple, efficient and reliable?." *in Proc. IEEE 24th international conference on software analysis, evolution and reengineering (SANER). IEEE, 2017.*

[105] G. Pinto and F. Castor. "Energy efficiency: a new concern for application software developers." *Communications of the ACM 60.12 (2017): 68-7*

[106] J. Hu and R. Marculescu. "Energy-and performance-aware mapping for regular NoC architectures." *IEEE Transactions on computer-aided design of integrated circuits and systems 24.4 (2005): 551-562.*

[107] C-L. Chou, U.Y. Ogras, and R. Marculescu. "Energy- and performance-aware incremental mapping for networks on chip with multiple voltage levels." *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 27.10 *2008: 1866-1879.*

[108] H. Khdr, et al. "Power density-aware resource management for heterogeneous tiled multicores." *IEEE Transactions on Computers 66.3 2016: 488-501.*

[109] A. Kumar, et al. "HybDTM: a coordinated hardware-software approach for dynamic thermal management." *in Proc. 43rd annual Design Automation Conference. ACM, 2006.*

[110] J. Choi, et al. "Thermal-aware task scheduling at the system software level." *in Proc. International Symposium on Low power electronics and design. ACM, 2007.*

[111] B. Salami, M. Baharani, and H. Noori. "Proactive task migration with a self-adjusting migration threshold for dynamic thermal management of multi-core processors." *The Journal of Supercomputing* 68.3, *2014: 1068-1087.*

[112] G. Bhat, et al. "Algorithmic optimization of thermal and power management for heterogeneous mobile platforms." *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 26.3, *2017: 544-557.*

[113] S. Liu, et al. "Hardware/software techniques for DRAM thermal management." *in Proc. IEEE 17th International Symposium on High Performance Computer Architecture. IEEE, 2011.*

[114] D.C. Snowdon, et al. "Koala: A platform for OS-level power management." *in Proc. 4th ACM European conference on Computer systems. ACM, 2009.*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MDAT.2020.2976669, IEEE Design and Test

[115] C. Xu, et al. "Automated os-level device runtime power management", *in Proc. ACM ASPLOS 2015.*

[116] H. Zhang and H. Hoffmann. "Maximizing performance under a power cap: A comparison of hardware, software, and hybrid techniques." *ACM SIGARCH Computer Architecture News* 44.2, 2016: 545-559.

[117] W. Seo, et al. "Big or little: A study of mobile interactive applications on an asymmetric multi-core platform." *in Proc. IEEE International Symposium on Workload Characterization. IEEE, 2015.*

[118] H.T. Dinh et al., "A survey of mobile cloud computing: architecture, applications, and approaches", *Wireless communications and mobile computing, vol. 13, 2013, pp. 1587-1611.*

[119] A. U. R. Khan et al., "A Survey of Mobile Cloud Computing Application Models," *in IEEE Communications Surveys & Tutorials, vol. 16, no. 1, pp. 393-413, 2014.*

[120] A. Khune, S. Pasricha, "Mobile Network-Aware Middleware Framework for Energy Efficient Cloud Offloading of Smartphone Applications", *IEEE Consumer Electronics, Vol. 8, Iss. 1, Jan 2019.*

[121] H. Li, K. Ota and M. Dong, "Learning IoT in Edge: Deep Learning for the Internet of Things with Edge Computing," *in IEEE Network, vol. 32, no. 1, pp. 96-101, Jan.-Feb. 2018.*

[122] A. Thomas et al. "Hierarchical and Distributed Machine Learning Inference Beyond the Edge," *in Proc. IEEE International Conference on Networking, Sensing and Control, May 9-11, 2019, pp. 18-23*

[123] J. Zhefeng, S. Mao: " Energy Delay Tradeoff in Cloud Offloading for Multi-Core Mobile Devices", *in IEEE Access, vol. 3, pp. 2306-2316, 2015*

[124] M. Shafi et al., "5G: A Tutorial Overview of Standards, Trials, Challenges, Deployment, and Practice," *IEEE Journal on Selected Areas in Communications, vol. 35, no. 6, pp. 1201-1221, June 2017.*

[125] D. Sabella, A. Vaillant, P. Kuure, U. Rauschenbach and F. Giust, "Mobile-Edge Computing Architecture: The role of MEC in the Internet of Things," *in IEEE Consumer Electronics Magazine, vol. 5, no. 4, pp. 84-91, Oct. 2016.*

[126] P. Mach and Z. Becvar, "Mobile Edge Computing: A Survey on Architecture and Computation Offloading," *in IEEE Communications Surveys & Tutorials, vol. 19, no. 3, pp. 1628-1656, 2017.*

[127] K. Zhang, et al. "Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks." *IEEE access 4 (2016): 5896-5907.*

[128] M. Jongerden and B. Haverkort, "Which battery model to use?", *IET Software, vol. 3, no. 6, 2009, pp 445-457.*

[129] T. Fuller, M. Doyle, J. Newman, "Simulation and optimization of the dual lithium ion insertion cell", *J. Electrochem. Soc., vol. 141, no. 1, 1994, pp. 1-10.*

[130] M. Doyle, T. Fuller, J. Newman, "Modeling of galvanostatic charge and discharge of the lithium/polymer/insertion cell", *J. Electrochem. Soc., vol. 140, no. 6, 1994, pp. 1526-1533.*

[131] J. Maxwell and J. McGowan, "Extension of the kinetic battery model for wind/hybrid power systems", *Proc. Fifth European Wind Energy Association Conf., 1994, pp. 284-289.*

[132] D. Rakhmatov, S. Vrudhula, D. Wallach, "Battery lifetime predictions for energy-aware computing", *Proc. ISPLED, 2002, pp. 154-159.*

[133] V. Rao, N. Navet, G. Singhal, A. Kumar, G. Visweswaran, "Battery Aware Dynamic scheduling for periodic task graphs", *Proc. IEEE International Parallel & Distributed Processing Symposium, 2006.*

[134] Y. Cai, S. Reddy, I. Pomeranz, B. Al-Hashimi, "Battery-aware dynamic voltage scaling in multiprocessor embedded syste"., *in Proc. ISCAS, 2005, pp. 616-619.*

[135] J. Cho, Y. Woo, S. Kim, E. Seo, "A battery lifetime guarantee scheme for selective applications in smart mobile devices", *IEEE Trans. on Consumer Electronics, vol. 60, no. 1, 2014, pp. 155-163.*

[136] D. Kadjo, R. Ayoub, M. Kishinevsky, P. Gratz, "A control-theoretic approach for energy efficient CPU-GPU subsystem in mobile platforms" *in Proc. ACM/IEEE Design Automation Conference, 2015.*

[137] S. Pourazarm and C. Cassandras, "Energy-based lifetime maximization and security of wireless-sensor networks with general nonideal battery models", *IEEE Trans. on control of network systems, vol. 4, no.2, 2017, pp. 323-335.*

[138] M. Gatzianas, L. Georgiadis, L. Tassiulas, "Control of wireless networks with rechargeable batteries", *IEEE Trans. on wireless communications, vol. 9, no. 2, 2010, pp. 581-593.*

[139] M. Schuchhardt, S. Jha, R. Ayoub, M. Kishinevsky, G. Memik, "CAPED: context-aware personalized display brightness for mobile devices", *in Proc. of ESWEEK-CASES, 2014.*

[140] B. Egilmez, M. Schuchhardt, G. Memik, R. Ayoub, N. Soundararajan, M. Kishinevsky, "User-aware frame rate management in Android Smartphones", *ACM Trans. On embedded computing systems -special issue ESWEEK, vol. 16, no. 5s, 2017.*

[141] E. Royraz and G. Memik, "Using built-in sensors to predict and utilize user satisfaction for CPU settings on smartphones", *Proc. ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, vol. 3, no. 1, 2019.*

[142] B. Egilmez, G. Memik, S. Ogrenci-Memik, O. Ergin, "User-specific skin temperature-aware DVFS for smartphones", *in Proc. DATE, 2015, pp. 1217-1220.*

[143] J. Park, S. Lee, H. Cha, "Accurate prediction of smartphones' skin temperature by considering exothermic components", *in Proc. DATE 2018, pp. 1500-1503.*

[144] D.N. Jha et al.: "IoTSim-Edge: A Simulation Framework for Modeling the Behaviour of IoT and Edge Computing Environments", 2019, [Online]: https://arxiv.org/abs/1910.03026

Sudeep Pasricha (sudeep@colostate.edu) received his Ph.D. in computer science from the University of California, Irvine in 2008. He is currently a Professor of ECE and CS at Colorado State University. His research interests include energy-efficiency and fault-tolerance for embedded and mobile computing. He is a Senior Member of the IEEE.

Raid Ayoub (raid.ayoub@intel.com) received the Ph.D. degree in computer engineering from the University of California, San Diego in 2011. He is a research scientist at the Intel Labs of Intel Corporation. His research interests include optimizations and dynamic control, high-level system modeling, and machine learning for emerging applications.

Sumit K. Mandal (skmandal@asu.edu) received his dual degree (BS + MS) from Indian Institute of Technology (IIT), Kharagpur. Currently, he is pursuing his Ph.D. in Arizona State University. His research interest includes analysis and design of NoC architecture, power management of multicore processors and AI hardware.

Umit Y. Ogras (umit@asu.edu) received his Ph.D. in electrical and computer Engineering from Carnegie Mellon University in 2007. He worked at Intel between 2008 and 2013. He is currently an Associate Professor at Arizona State University. His research interests include embedded systems, wearable internet-of-things, flexible hybrid electronics, and mobile platforms.

Michael Kishinevsky (michael.kishinevsky@intel.com) leads a system design and architecture group at Strategic CAD Labs of Intel Corporation. His research interests include system-level design and optimization, and communication networks. He received his PhD in computer science from the Electrotechnical University of St. Petersburg, Russia. He is a senior member of IEEE.

Mail Address: 1373 Campus Delivery, Colorado State University, Fort Collins, CO 80523-1373.