Energy Consumption in Mobile Devices: Why Future Systems Need Requirements-Aware Energy Scale-Down

Robert N. Mayo and Parthasarathy Ranganathan
Hewlett Packard Labs
1501 Page Mill road MS 1177
Palo Alto California 94304
{Bob.Mayo,Partha.Ranganathan}@hp.com

ABSTRACT

The current proliferation of mobile devices has resulted in a large diversity of designs, each optimized for a specific application, form-factor, battery life, and functionality (e.g., cell phone, pager, MP3 player, PDA, laptop). Recent trends, motivated by user preferences towards carrying less, have focused on integrating these different applications in a single general-purpose device, often resulting in much higher energy consumption and consequently much reduced battery life. This paper argues that in order to achieve longer battery life, such systems should be designed to include requirements-aware energy scale-down techniques*. Such techniques would allow a general-purpose device to use hardware mechanisms and software policies to adapt energy use to the user's requirements for the task at hand, potentially approaching the low energy use of a special-purpose device.*

We make two main contributions. We first provide a model for energy scale-down. We argue that one approach to design scale-down is to use special-purpose devices as examples of power-efficient design points, and structure adaptivity using insights from these design points. To understand the magnitude of the potential benefits, we present an energy comparison of a wide spectrum of mobile devices (to the best of our knowledge, the first study to do so). A comparison of these devices with general-purpose systems helps us identify scale-down opportunities. Based on these insights, we propose and evaluate three specific requirements-aware energy scale-down optimizations, in the context of the display, wireless, and CPU components of the system. Our optimizations reduce the energy consumption of their targeted subsystems by factors of 2 to 10 demonstrating the importance of energy scale-down in future designs.

1. Introduction

Recent advances in computing and communication have led to increased use of a large number of mobile computing devices. These devices have many purposes and form-factors including both general-purpose devices like laptops, pocket PCs, and palm computers as well as specialized devices like portable MP3 players and e-mail pagers. Both form factor and energy are critical resources for all these devices, forcing users to trade away functionality to gain smaller form factors and longer battery lifetimes. A variety of such tradeoffs exist for many specific tasks. For instance, the email task can be accomplished with a feature-rich and high-energy application like Microsoft Outlook running on a laptop, or a reduced-feature low-energy application like a Blackberry email pager[1].

The drive for small form factors is strong, resulting in users demanding the most value from a given form factor. The popularity of camera/cell phone combinations are examples of multiple applications in a single device, with a consequent reduction in overall form factor when compared to the sum of two separate devices. Most successfully-converged products combine applications with similar hardware and software requirements. In our example, both cameras and cell phones can share a processor with limited computational power, a common display, and a small set of input buttons. But different applications inherently have some mismatch. Higher-resolution color displays are needed on camera/phone combinations compared to stand-alone mobile phones, leading to increased energy use even when only using the phone portion of the device.

Perhaps the best examples of this mismatch are in general-purpose devices like PDAs. These devices can run a wide range of software and accept a wide range of hardware cards. Thus, they would appear to be the ultimate device for handheld convergence. But this generality currently comes at the cost of high energy use, making the devices much less attractive. The main reason for this is the lack of adaptability in the hardware and software energy use.

Consider a comparison of two email applications: feature-rich email software running on a high-end handheld computer versus an email pager. Both handle the email task well, but with different tradeoffs between functionality and battery life. In

---

[1] In this paper, we use the terminology *task* to mean a broad category, such a music listening task or an email task. A specific set of hardware and software to accomplish this task is called an *application*, and involves various tradeoffs of functionality, battery life, and form factor. For instance, the music listening task could be accomplished using either a PDA or a tiny MP3 player.

the case of the pager, users want long battery life, notification of incoming email, and an acceptable screen for text. Much less important is the ability to view color photographs or read attached files. If the user of a handheld computer desires the reduced features and longer battery life, it is largely unobtainable. It simply is not possible to run a pager-like application on a high end handheld computer and get pager-like battery life.

Application-specific devices like cell phones, email pagers, and MP3 players are examples of highly successful tradeoffs of functionality, form-factor, and battery life. Their success proves them to be excellent points in the design space, in that users find high value in them. As such, they can serve as benchmarks against which we can compare general-purpose devices. A truly general-purpose device should strive to emulate these design points, including not only the specific functionality but also the battery life.

While there are many technical challenges, *requirements-aware energy scale-down* is an approach that can yield improvements. The idea is to have both hardware and software that can scale their features and energy use to meet a variety of design points. For high functionality, hardware and software would present a rich set of features to the user, at the cost of higher energy use. At the low end, hardware and software would scale their functionality to match a popular low-end design point, resulting in lower energy use. Specifically, we recommend considering each component in the general-purpose device and comparing it to the requirements of the applications using that device. Ideally, each general-purpose component should be capable of scaling down its energy use to match the design point used by the application with the lowest requirements. There are two alternatives for achieving this: *gradation-based scaling*, where the component itself has a wide range of adaptability, or *plurality-based scaling*, where the device chooses among multiple components with different properties. We suggest that a good way of determining mismatches in the component's functionality and the task requirements for a general-purpose device is to use as reference the component's functionality in a special-purpose device targeted at that application.

The rest of the paper is organized as follows. The next section of the paper (Section 2) further discusses the need for and potential from energy scale-down. In particular, it characterizes the energy costs of convergence and illustrates how the design of special-purpose devices can help identify energy scale-down methods in general-purpose devices. As a way of validating our energy scale-down approach, Section 3 proposes three specific scale-down optimizations in the context of the display, wireless, and processor components of the system and shows how they can significantly enhance battery life. Section 4 concludes the paper.

2. Energy Scale-Down

2.1 Energy Costs of Convergence

To validate the hypothesis that special-purpose devices are low-energy devices when compared with converged or general-purpose devices, we measured the energy use of a broad range of devices used for different tasks.

2.1.1 Methodology

We are not aware of any other previous work that has performed such a broad comparison of mobile devices for such a wide range of mobile tasks. Consequently, we have had to make a number of choices when designing our experimental methodology. Of the large design space possible for a study like ours, we chose to focus on quantifying the wide range of energy usage *per task*. We chose our tasks to be representative of the typical activities mobile users would perform. We focused on commercial products optimized for one or more of these tasks.

Devices: Figure 1 summarizes the key characteristics of the devices that we use. The individual devices we consider include a laptop, a handheld, a cell phone, a high end pager, a high-end MP3 player, a low-end MP3 player, and a small "memo" voice recorder. Unless otherwise noted, all the units were set to default settings. The laptop and handheld were set to never turn off. Also, given that the backlight for the handheld chosen is relatively unique in the handheld market (with power consumption atypical of other handhelds in the market), we performed all our experiments with the backlight set to minimum power mode.

Class: Device	CPU	Storage	Display	Wireless	Interfaces	OS
Laptop: Compaq Armada M300	600 MHz Pentium II	256 MB RAM, 12 GB disk	1024x768, 12.1" TFT	Lucent WaveLAN Gold PCMCIA, IEEE 802.11	full-function keyboard, speaker, mic., stereo audio-output jack, and other additional general purpose functionality	Win XP Pro
Handheld: Compaq iPAQ 3630	206 MHz Intel Strong ARM	32 MB RAM, 32 MB ROM	240x320 2.26" TFT display	Same as laptop	touchscreen interface, speaker, microphone, stereo audio-out jack	Pocket PC
Cell phone: Nokia 8260	not in spec	not in spec	73x50, 1.2"x0.8" monochrome LCD+backlight	AT&T wireless	GSM-like headset jack, vibration notification and audio output	Proprietary
High-end pager: Blackberry W1000	Intel 386 (speed not in spec)	4 MB flash, 512 KB SRAM	8-line (x28 char) LCD+backlight	Tx frequency:896-902 MHz; Rx freq: 935-941MHz	Trackwheel, 31-key qwerty keypad, tone and vibration notification	Proprietary
Low-end MP3: Compaq iPAQ PA-1	not in spec	two 32MB flash card	7x66, 1"x0.4" LCD +backlight	None	Buttons	Proprietary
High-end MP3: Nomad jukebox (DAP 6G01)	not in spec	8MB DRAM, 6GB disk	132x64 LCD+backlight	None	Stereo headphone jack	Proprietary
Voice recorder: VoiceIt VT-90	not in spec	not spec (max. record time of 90 seconds)	None	None	Buttons	Proprietary

Figure 1: Devices evaluated in power comparison study

Task	Description
Email	The benchmark tries to capture typical activities associated with an email application. The first component (Rcv) captures the power for receiving messages and the power for the notification events. An automatic script from a remote machine sends out two sets of 10 messages separated by a pause. All volumes are set to maximum and vibrate-mode, if any, is turned on. The second benchmark (Reply) includes aspects of reading, composing, and sending messages.The benchmark models seven forwards and one single-line reply. The messages chosen include a 10KB HTML announcement and a 4KB text message.
MP3	This benchmark measures the power consumed to play the first two minutes of a 6.44MB MP3 song recorded at a bit rate of 192 Kbps. The default Windows Media Player was used to play the song on the laptop and handheld; the high-end and low-end MP3 players had proprietary interfaces to play the song. Power readings were taken with the same set of headphones for all the appliances. Additionally, when available, the power was also measured with the speaker (for the laptop and the handheld). In these cases, the power was measured with the volume set to maximum.
Web browsing	The web browsing benchmark included connecting to an external link with a large number of embedded images. The benchmark refreshes the page once the page is downloaded. All experiments were performed at similar times to minimize network effects.
Text notes taking	This benchmark included the first two minutes of typing in the rules from the table of contents of "The Elements of Style" by Strunk and White, by an operator familiar with the user interface.
Audio notes taking	For this benchmark, we read out loud the first 9 rules from the table of contents of the same book as above one at a time and played back the recordings one at a time.
Two-way messaging	We designed two benchmarks to capture the activities associated with messaging for a mobile user. The first benchmark stresses *instant text messaging*, while the second benchmark stresses *voice chats (or phone conversations)*. In both these cases, we followed a pre-determined script. In order to capture the power consumption of notification events (audio or vibration alerts), we began the conversation with a request into the measured device and then one minute into the messaging, we disconnected the conversation and began another conversation, this time initiated by the measured device. As before, we assumed reasonable skills with the user interfaces, and all volume controls were set to maximum. For the windows environments (laptop and handheld), we used MSN messenger for the text and voice chats.

Figure 2: Tasks evaluated in power comparison study

Tasks: Figure 2 summarizes the key tasks that we studied. The tasks we consider include email handling (notification, sending and receiving), MP3 song-playing (speaker and headphone), web browsing, notes taking (text and voice), and two-way messaging (text and voice). For each application (a task/device combination), we designed a two-minute long benchmark that we felt represented typical use. In applications where there were elements of non-repeatability, we repeated the experiments several times to ensure that all effects were adequately captured

.

	Email		MP3			Notes		Messaging		
Device	**Rcv**	**Reply**	**Speaker**	**Headphone**	**Browse**	**Text**	**Audio**	**Text**	**Audio**	**Idle**
Laptop	15.16 W	16.25 W	18.02 W	15.99 W	16.55 W	14.20 W	14.65 W	14.40 W	15.50 W	13.975 W
Handheld	1.386 W	1.439 W	2.091 W	1.700 W	1.742 W	1.276 W	1.557 W	1.319 W	-	1.2584 W
Cellphone	539 mW	472 mW	-	-	-	-	-	392 mW	1147 mW	26 mW
Email Pager	92 mW	72 mW	-	-	-	78 mW	-	-	-	13 mW
High-end MP3	-	-	-	2.977 W	-	-	-	-	-	1.884 W
Low-end MP3	-	-	-	327 mW	-	-	-	-	-	143 mW
Voice Recorder	-	-	-	-	-	-	166 mW	-	-	17 mW
variance	*16496%*	*22727%*	*861%*	*4890%*	*950%*	*18252%*	*8825%*	*3673%*	*1351%*	*107500%*

Figure 3: Power consumption for special-purpose and general-purpose mobile devices

2.1.2 Measurement setup

For our experiments we measured total system power. We measured the current drawn by the device by measuring the voltage across a 0.10 ohm sense resistor (tolerance 1%) between the power source and the device itself. In order to reduce noise, we amplified the voltage across the resistor with a Maxim MAX4172 precision high-side current-sense amplifier, which was then measured by a data acquisition system. For all devices except the cell phone, we removed the batteries and ran the device from a DC power supply, measuring current as it entered the device. Since our cell phone wouldn't run in this manner, we used a fully charged battery with a sense resistor inserted between the chemical cell and the additional electronics we found within the battery case. In order to observe time-varying behavior, we collected a 2 minute trace of each device and application at sample rate of 10,000 samples per second. Traces included both the current (as output by the MAX4127) as well as the power supply voltage of the system under test. Power was computed as the product of the voltage and current samples. In this paper, we report only average numbers in the interests of space.

2.1.3 Measurement Results

Figure 3 summarizes the average power consumption of the devices and benchmarks that we consider. As we expect, we observe that the different devices spend different amounts of power, even when providing similar service. However, more surprising, we see that the variations between these readings are very large, ranging from 950% to over 22,000%. For example, the energy use of our MP3 application varies by a factor of 49 when playing the identical music on different devices. Our email reply benchmark consumes between 71.5mW and 16.26W on different devices; that corresponds to nearly a factor of 227.

2.2 The Need for Energy Scale-Down

In all the cases in the previous section, the large variances were primarily attributable to the difference between the low energy consumed by an application-specific device optimized for energy and the high-energy consumed by a general-purpose device optimized for functionality. The energy differences can be largely explained in terms of the individual components in each system. Focusing on one specific example of the email application, when moving from a laptop (highest power) to an email pager (lowest power), a number of components are replaced with less powerful components. This provides lower power and lower, or perhaps different, functionality (but optimized for the characteristics of the application and market acceptance of reduced features). The display and CPU are scaled down, and the wireless system has different characteristics. The application software is scaled down to provide just the essential features.

The scale-down of the software is particularly interesting, since it often tends to be ignored. In our example, the software in these benchmarks includes Outlook in the laptop, Pocket Outlook on the iPAQ, and simple text email software on the cell phone and the pager. Each of these provides different functionality. Unlike a study of, say, CPU performance, where we would like to keep the benchmarks constant to ensure a fair comparison, we argue that when comparing power consumption of different implementations of tasks, software is an important component that also needs to be scaled to meet the user' s desired functionality.

Below, we discuss the power numbers and how they relate to our hypothesis that devices with general-purpose or combined functionality consume more power because they do not provide the adaptivity to respond to application requirements. In these discussions, we highlight examples of how the optimizations found in special-purpose devices can be useful in improving energy efficiency for our general-purpose devices, in the context of one application - email.

Email application on different mobile devices: Comparing the cell phone and the email pager device for the email benchmark, we observe an interesting trend with the pager having a factor of 6 lower power in spite of its larger (and hence potentially higher power) display. An examination of the traces indicates that that the pager's wireless system has significantly lower activity compared to the cell phone. The cell phone demonstrates the compromises of convergence, even on small special-purpose devices. Like other handheld devices, the wireless protocol for a cell phone device typically leaves the radio off most of the time, turning it on periodically to check for activity. Our traces of both the phone and pager show periodic energy spikes that we hypothesize are the radio waking up. On the phone, these spikes are approximately 1.2 seconds apart. This corresponds to adding an average ring latency of 0.6 seconds to each incoming phone call. On the pager, these spikes are approximately 5 seconds apart, leading to an average 2.5 second latency on incoming email (a factor of 4 compared to the cell phone). Both these latency numbers seem appropriate to the application at hand, and this variation in the wakeup period can be considered another example of component scale-down for better energy efficiency. Thinking in terms of scale-down, we might consider designing a cell phone protocol in a way that allows us to set the phone to an email-only mode, allowing us to lengthen the average latency to 2.5 seconds, approaching the energy consumption of an email pager.

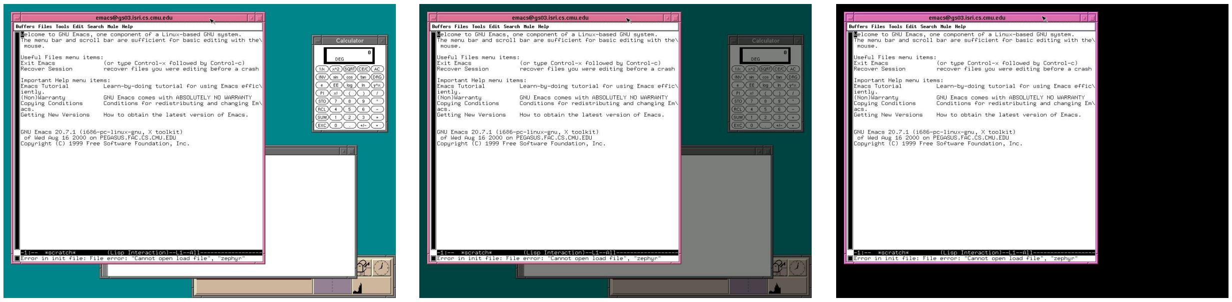
Comparing the cell phone against the handheld shows an additional energy cost for additional generality. The handheld email benchmark consumes approximately 3 times the energy as the cell phone. The wireless system for the handheld, in particular, is optimized for low latency local communication at a high bit rate, rather than wide-area communication at a bit-rate tuned for speech. The ideal scalable handheld would incorporate both types of wireless systems or (in an ideal world) a single system that could scale its features to match either wireless system. Even without changing the range or bandwidth, however, one scale-down approach would be to change the latency requirements of the wireless system to match the task at hand. A typical wakeup period for 801.11b is 100ms, while 5 seconds would be fine for email. Even though the 802.11b protocol supports longer wakeup periods, software generally does not provide the necessary interfaces for taking advantage of this. A software system designed for scale-down of energy would provide these interfaces, and applications would facilitate scaling the components to match the requirements of the task at hand.

In addition to the wireless component, we can observe opportunities for scale down in other components of the system. One important component of the handheld is the display which has no scale-down capacity. Even if the users are comfortable with a smaller, lower-resolution, lower-color screen to scroll through when reading their messages, there are no software or hardware interfaces to support this. This problem is particularly exacerbated in the laptop with its emphasis on a much larger form-factor. Similarly, the processor component of the various devices varies significantly in power. The rated power for the Pentium II and StrongARM processors used in the general-purpose device is orders of magnitude higher than the rated power of the embedded processors used in the special-purpose devices, even though for some of our tasks, similar computation is performed on all the processors.

Summary: Though we focused on one application in the interests of space, similar trends are present in the other applications as well. For all the tasks, our results show that the special-purpose devices have orders of magnitude lower power consumption compared to the general-purpose devices, validating our intuition that they could serve as good examples of energy use that general-purpose devices may aim for. We suggest that researchers, in part, evaluate the success of their energy scale down efforts by how closely they approach the energy consumption of such application specific devices. For example, in our ideal world, a laptop playing an MP3 file could, if the user desired, consume no more power than the best available MP3 player. The next section discusses specific energy scale-down optimizations that work towards this goal.

3. Energy Scale-down Optimizations

As a way of closing the gap between the special-purpose and general-purpose devices, we suggest integrating the notion of *requirement-specific energy scale-down* at all levels of the system, namely providing for the *design and use of adaptivity in hardware and software* to exploit mismatches between system functionality and workload/user requirements. Specifically, we suggest considering each component in the general-purpose device and comparing it to the requirements of the applications using that device. Ideally, each general-purpose component should be capable of scaling down its



```
Welcome to GNU Emacs, one component of a Linux-based GNU system.
The menu bar and scroll bar are sufficient for basic editing with the\
 mouse.

Useful Files menu items:
Exit Emacs              (or type Control-x followed by Control-c)
Recover Session         recover files you were editing before a crash

Important Help menu items:
Emacs Tutorial          Learn-by-doing tutorial for using Emacs effic\
iently.
(Non)Warranty           GNU Emacs comes with ABSOLUTELY NO WARRANTY
Copying Conditions      Conditions for redistributing and changing Em\
acs.
Getting New Versions    How to obtain the latest version of Emacs.


GNU Emacs 20.7.1 (i686-pc-linux-gnu, X toolkit)
 of Wed Aug 16 2000 on PEGASUS.FAC.CS.CMU.EDU
Copyright (C) 1999 Free Software Foundation, Inc.

-1:--  *scratch*         (Lisp Interaction)--L1--All------------------
Error in init file: File error: "Cannot open load file", "zephyr"
```

Original Interface Background half dim Background fully dim

Figure 4: Display interfaces with energy scale-down

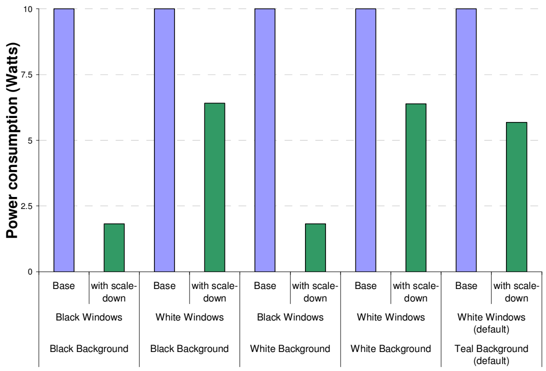


Figure 5: Display scale-down benefits

energy use to match the design point used by the application with the lowest requirements. There are two alternatives for achieving this: *gradation-based scaling*, where the component has a wide range of adaptability, or *plurality-based scaling*, where the device chooses among multiple components with different properties. Below, we discuss three example scale-down optimizations in the context of the display, wireless, and processor components of the system that attempt to use adaptivity to improve the efficiency of energy use in the device.

3.1 Display Scale-Down

The user acceptance of smaller, lower-quality, and lower-energy displays in special-purpose devices indicates that certain tasks may not always need the most aggressive functionalities of the display (e.g., large size, full color, great resolution, backlight, etc.). In contrast, most current general-purpose systems include *"one-size-fits-all"* displays targeted at the needs of the most aggressive workload/user. This can lead to large energy-inefficiencies in the display energy consumption of other workloads and users. This motivates the need for *energy-adaptive display systems* [IyerLuo+2003] that consume energy only on equivalently displayed, with no loss in visual quality, on much simpler lower power displays. Our analysis of the user traces indicated that many of these mismatches could be traced back to the typical content of the windows as opposed to specific user preferences. For example, windows with low content (email composition, terminals, system status and control messages, menu widgets, etc.) were the dominant types of smaller-sized windows and windows with relatively higher content (web browsing, code development, PowerPoint, document reading, etc.) were the dominant types of larger-sized windows.

Based on these observations, we evaluated a family of display scale-down optimizations built on emerging OLED display technologies [Stanford2001] that allow the energy consumption to be proportional to the overall light output of the display. At a software level, we designed *energy-aware user interfaces* that change the luminescence and color of the non-active screen areas to reduce power while leaving the active screen area (the window of focus) unchanged. Some examples of the interfaces presented to the user are summarized in Figure 4. Our experience with prototypes indicates broad acceptance of these interfaces among users, particularly in the context of longer battery life. Figure 5 summarizes the energy benefits from applying the fully-dimmed optimization. Since the energy benefits are a function of the screen background and the window background colors, in addition to the default windows configuration (teal screen background and white windows background) we bracketed our results by evaluating other configurations. As the results indicate, integrating scale-down optimizations in the display design can achieve factors of 1.5-5 reduction in the display energy. The benefits for individual users vary, up to 10X in some cases.

In addition to the specific family of designs that we evaluated, energy scale-down can be integrated in other ways as well. For example, with display

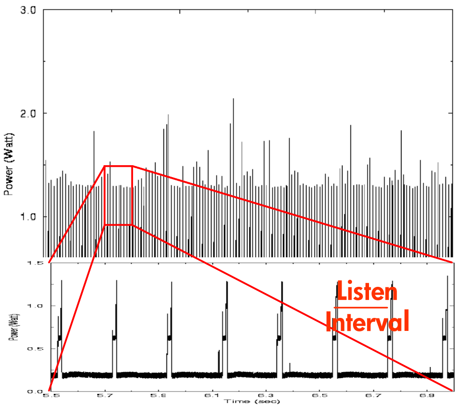


Figure 6: Wireless power consumption from beacons

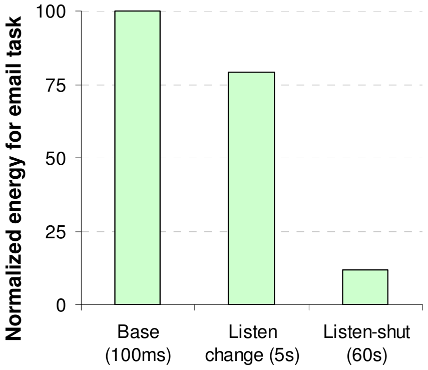


Figure 7: Wireless scale-down benefits

technologies that do not support energy variability, a hierarchy of displays or alternative communication and user input methods could be used to provide the hardware support for adaptivity. Similarly at the software level, we can extend the optimizations discussed above to include pointer-based user-relevance determination, time-based dimming interfaces, as well as intra-application support for adaptivity. Though not presented here in detail, the power benefits and user acceptance of energy-adaptive display interfaces for handhelds was also studied. Again, the results show factors of 1.3 to 8.3 with high user acceptance ratings [Harter+2003].

3.2 Wireless Scale-Down

As discussed earlier, special-purpose communication devices such as cell phones or pagers consume significantly lower energy in the wireless system by minimizing the activity on the wireless network. In contrast, the general-purpose systems often have wireless subsystems and protocols optimized for the most aggressive connection requirements (e.g., bandwidth, latency, response) leading to energy inefficiencies. The wireless scale-down optimization discussed in this section addresses this drawback.

To understand the workload's requirements for the wireless system, we evaluated the energy spent in the 802.11b wireless sub-system of a general-purpose mobile device for an email application [AbouGhazalaMayo+2003]. Our results indicated that a large fraction of the wireless energy was spent when the system was in idle mode (as opposed to transmitting/receiving messages). Further, as indicated in Figure 6, the power consumed in the idle mode was dominated by the power spent in listening for periodic beacons to ensure timely response to incoming transmissions. Typical default configurations set the "listen interval" (the time between beacons) to be 100 ms. In contrast, the mean time period between message receipts for our representative user was many minutes.

Based on these observations, we implemented an energy-adaptive wireless system that could adapt its listen interval to better respond to the desired application response times. For example, if the user finds an email latency of 5 seconds to be acceptable, this information should be reflected in the listen interval of the wireless protocol. We evaluated two approaches. For short increases in listen interval (5 seconds, *"Listen change"* in Figure 7), we changed the listen interval by changing the corresponding parameter in the 802.11 protocol. For longer increases (60 seconds, *"Listen shut"* in Figure 7), we turned off and restarted the wireless card. As shown in the Figure, the wireless scale-down optimizations achieve factors of 1.3 to 9 better energy consumption.

Though the specific optimization considered above is fairly straightforward, the same insights can be applied to other configurations, particularly in the context of multiple wireless networks in the same device. In particular, a small amount of additional hardware in the form of a small low-power radio to supplement the main wireless network can provide even finer-granularity of adaptivity [ShihBahl+2002].

3.3 Processor Scale-Down

The third scale-down optimization that we consider focuses on the processor component of the system power and is motivated by the observation that in some cases, a lower power and lower functionality processor is often enough to adequately perform a particular task. Once again, the general-purpose system includes a processor that is typically targeted at the most aggressive workload requirements (performance) and does not have a simple mechanism to scale down to the lower functionalities required by other tasks.

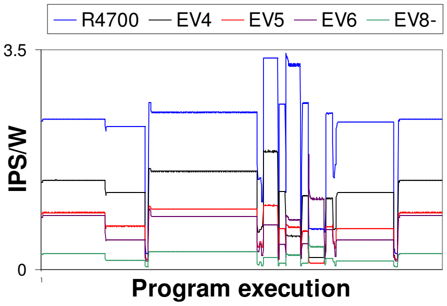


Figure 8: Energy efficiency differences between processor cores

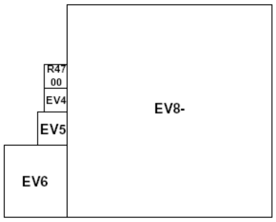


Figure 9: CPU scale-down architecture

Fourteen integer and floating-point applications from the SPEC2000 benchmark suite were simulated on five different processor cores supporting the same instruction set architecture [KumarFarkas+2003] [KumarFarkas+2003b]. Figure 8 shows the energy efficiency of the five cores over the course of execution of a representative benchmark. The five cores approximate the MIPS R4700, the Alpha 20164 (EV4), 21164 (EV5), 21264 (EV6) and a potential next-generation approximation to the EV6 (EV8-). The results indicated that different processor cores have different energy efficiencies based on the nature of the workload being executed on them.

Based on the analysis, a plurality-based scale-down optimization for processors based on heterogeneous single-ISA multi-core architectures may yield energy benefits. The key idea is to have the main high-performance processor supplemented with other satellite processors that span the power-performance design space. The workload is run on the core with the best energy efficiency properties for that workload, and the other cores are shut off. This should be possible with little additional die area, as the die size of such a combined system is little more than the size of the largest processor, as shown in Figure 9. Evaluation of both static and dynamic heuristics for workload migration indicate an average energy improvement of 1.4X (factors of 2 to 10 in six of the applications) with less than 3% speed degradation. In cases when lower performance is acceptable, it is possible to achieve factors of 3 to 11 reduction in energy with less than 25% loss in performance[KumarFarkas+2003][KumarFarkas+2003b].

Though limited to the specific technology implementation of the processor, another potential scale-down optimization for processors is the use of voltage and frequency scaling [PillaiShin2001]. Additional scale-down optimizations can also consider the use of energy-aligned cores that design the architecture of the specific cores to better improve energy efficiency.

4. Conclusions

As the mobile device market matures, the large number of mobile computing devices optimized for different form factors and functionalities is likely to be replaced with a few commonly-accepted devices that integrate multiple functionalities in the same device. Indeed, this is evidenced by the large number of announcements about "combination products" such as camera/cell phones or cell phones/personal organizers or gaming devices/MP3 players, etc. However, while there has been a great focus on designing these devices to scale up the device properties to provide the greatest of the various functionalities, there has been very little work in providing approaches to scale-down the device to the least of the various functionalities. This has particularly been a problem in the case of the energy consumption of these devices when a system component consumes more energy by virtue of supporting a larger function set than what is desired by an application (E.g., reading black and white text messages on a color screen cell phone organizer).

This paper argues that along with the importance given to scaling up functionality, equal importance should be given to designing methods in hardware and software to *scale down* the energy. Individual applications or users can then use these mechanisms to control the energy based on their specific requirements. As validation of this thesis, we compare the energy consumption of general-purpose devices (that support the function set required by several tasks) with special-purpose devices targeted specifically at particular tasks. Across the range of tasks we considered – sending and receiving email, web browsing, listening to MP3 music, text and audio notes taking, and text and phone messaging – we observed inefficiencies in the general-purpose devices that led to factors of 10 to 100 higher energy consumption compared to the special-purpose devices. To the best of our knowledge, ours is the first such study to perform a consistent comparison of the energy consumption of the various devices. Furthermore, an analysis of the differences between the devices illustrates opportunities when user requirements can be met with much lower energy use.

Building on this analysis, we proposed and evaluated three specific scale-down approaches that exploit an awareness of the user and task requirements to scale down the energy selectively. In the first case, the system leverages the observation that users typically use only a fraction of their screen area and selectively controls the pixel intensity on the screen to match the power consumed in the display with the portions relevant to the user. In the second case, the system leverages the observation that users are willing to tolerate longer response times than what is currently provided by wireless networks and by exposing this tolerance to the protocol, reduces the power consumed in the wireless system. The third case observes that different processor designs are better matched, from an energy efficiency point of view, to different workloads and uses a multi-core architecture to reduce energy. In all these cases, the energy scale-down optimizations achieves close to a factor of 2 to 10 better energy consumption compared to existing methods of designing systems.

While the three optimizations we consider in the paper validate our argument on the potential of energy scale down in future designs, we believe that we have only scratched the surface. Previously proposed means for adaptivity can complement these to provide further scale-down, for example, voltage and frequency scaling [PillaiShin2001], architectural gating [ManneKlauser+1998], selective memory usage [LebeckFan+2000] and disk spin-down [DouglisKrishnan+1994]. However, the factors of 10 to 100 indicated in our energy comparison show that we still have a significant potential in terms of energy savings to attain. Additional mechanisms for adaptivity in hardware and software for energy scale down and policies for requirements-aware use of this adaptivity will be essential as we try to further address the battery life challenges in future systems.

5. References

[AbouGhazalaMayo+2003] Nevine Abou-Ghazala, Robert Mayo, and Parthasarathy Ranganathan. Idle Mode Power Management for Personal Wireless Devices, *Hewlett Packard Technical Report HPL 2003-102*, 2003

[DouglisKrishnan+1994] Fred Douglis, P. Krishnan and B. Marsh. Thwarting the power hungry disk. *Proceedings of the 1994 USENIX Conference*, pp 293-306, January 1994

[Harter+2003] Timoth Harter, Sander Vroegindeweij, Erik Geelhoed, Meera Manahan, and Parthasarathy Ranganathan. The Potential for Energy-aware User Interfaces on Handheld Designs. *Hewlett Packard Technical Report HPL 2003-*, 2003

[IyerLuo+2003] Subu Iyer, Annie Luo, Robert Mayo, and Parthasarathy Ranganathan. Energy-Adaptive Display System Designs for Future Mobile Environments. In *Proceedings of the First International Conference on Mobile Systems, Applications, and Services*, 2003, pp245-258

[KumarFarkas+2003] Rakesh Kumar, Keith Farkas, Norman Jouppi, Parthasarathy Ranganathan, and Dean Tullsen. Processor Power Reduction via Single-ISA Heterogeneous Multi-core Architectures. *Computer Architecture Letters*, Volume 2, April 2003

[KumarFarkas+2003b] Rakesh Kumar, Keith Farkas, Norman Jouppi, Parthasarathy Ranganathan, and Dean Tullsen. Single-ISA Heterogeneous Multi-Core Architectures: The Potential for Processor Power Reduction, *To appear in the Proceedings of the 36th International Symposium on Microarchitecture, December 2003*

[LebeckFan+2000] A. Lebeck, X. Fan, H. Zeng, and C.S.Ellis. Power aware page allocation. In *Proceedings of the Ninth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS IX)*, November 2000

[ManneKlauser+1998] S. Manne, A. Klauser, and D. Grunwald. Pipeline gating: Speculation control for energy reduction. In *Proceedings of the 25th Annual International Symposium on Computer Architecture*, pp 132-141, June 1998

[PillaiShin2001] P. Pillai and K. G. Shin. Real-time dynamic voltage scaling for low-power embedded operating systems. In *Proceedings of the 18th Symposium on Operating System Principles*. 2001.

[ShihBahl+2002] Eugene Shih, Paramvir Bahl, and Michael J. Sinclair. Wake on Wireless: An Event Driven Energy Saving Strategy for Battery Operated Devices. In *Proceedings of the Eighth Annual ACM Conference on Mobile Computing and Networking*, Atlanta, Georgia, USA, September 2002.

[Stanford2001] Stanford Resources Inc. *Organic Light Emitting Diode Displays: Annual Display Industry Report*, Second Edition 2001