Display Power Management Policies in Practice

Stephen P. Tarzia*   Peter A. Dinda*   Robert P. Dick†   Gokhan Memik*

*Northwestern University
EECS Department
Evanston, IL, USA

†University of Michigan
EECS Department
Ann Arbor, MI, USA

{spt175,pdinda,memik}@eecs.northwestern.edu   dickrp@eecs.umich.edu

ABSTRACT

We present the first study of the real-world behavior of display power management (DPM) policies. DPM policies control the mechanism of powering on and off the display—turning off the display typically reduces total system power by ∼31%. The most widely used DPM policy, *human interface device (HID) timeout*, powers off the display after a user-configurable period of human interface device inactivity, and powers it back on in response to activity. To increase energy savings, we also consider an alternative policy, *user presence detection*, that uses sonar sensing to power off the display when user absence is detected. Our study captures how these DPM policies work "in the wild", both in terms of energy savings and the user irritation. We also determine the maximum energy saving opportunity for any DPM policy, based on measured behavior. Our study, based on a 3,738 hours of computer usage by 181 volunteers with different machines, reveals several surprising results. User idle periods follow power law distributions with little temporal correlation. The maximum possible reduction in energy used for the display is 81%, while the HID timeout policy manages to reduce this energy by 51%. Many users have already customized the HID timeout policy on their machines, resulting in a high variation of timeout values, and surprisingly low levels of user irritation. However, the 44% of users that have not customized HID timeouts experience more irritation. The proposed user presence detection policy, when effective, further reduces display energy consumption by 10% when combined with the HID timeout policy. 40% of the 2,869 machines tested can effectively generate and record ultrasound for sonar.

Categories and Subject Descriptors

B.4.2 [**Input/Output and Data Communications**]: Input/Output Devices—*Image Display*; D.4.m [**Operating Systems**]: Miscellaneous

General Terms

Measurement, Human Factors, Performance

1. INTRODUCTION

Display power management (DPM) policies are autonomic control policies with which all computer users have direct experience. The goal of such a policy is to turn off the computer display whenever the user becomes inattentive and turn it back on again when the user becomes attentive. Toggling the display is a simple, but important, mechanism for controlling energy consumption. By display, we mean the LCD panel and its backlight (this paper does not consider the GPU).

LCD displays are major contributors to energy consumption. A past power measurement of laptops indicates that when dynamic voltage and frequency scaling (DVFS) is enabled on the CPU and the system is idle, the LCD (9%) and backlight (29%) together draw 38% of system power when fully brightened [10]. This idle condition is an extremely common state for modern laptop computers [12, 4]. We present our own measurements of the potential energy savings (which exhibit similar behavior) in Section 3.

The DPM policy determines when to toggle on or off the display. The resulting energy savings are determined by user behavior, i.e., attentive and inattentive periods, and the effectiveness of the policy. Note that user attention is not usually precisely measured although a strictly optimal policy would require this information. There is an important tension in all realizable DPM policies: the more aggressive the DPM policy is in estimating user inattentiveness, the higher the energy savings, but also the higher the chance of annoyingly shutting off the display when the user is still attentive (e.g., reading a web page). We call such policy errors *irritation events*. A DPM policy must choose an operating point to resolve the tension between energy savings and irritation. In this work, we analyze how efficiently existing schemes make such trade-offs. Specifically, we answer the following questions:

- How well do current DPM policies work, both in terms of energy savings and user satisfaction?
- How much better could an optimal DPM policy do? How close to optimal are current policies?
- How can current DPM policies be improved upon?

We address these questions by examining the measured results of a large real-world study of two policies. The policies are the *human interface device (HID) timeout* policy that

is universally employed in current operating systems, and a policy we developed, *user presence detection*, that tries to improve upon HID timeout, and that could be used in conjunction with it. We describe these policies in detail in Section 4.

Our measurements are based on a large-scale user study of 181 volunteers, each with a separate machine, who collectively provide us with 3,738 hours of computer usage logs. We describe our study in detail in Section 6. We are aware of no prior work that evaluates any DPM policies in practice: this study is our main contribution.

In summary, our data show that:

- In practice, DPM policies can reduce display energy consumption by 81% at most and total system energy consumption by 25%.
- The widely-used HID timeout policy is surprisingly effective in practice, reducing display energy consumption by 51% and total system energy consumption by 15.3%. Furthermore, the median rate of irritation events for this policy is low: on the order of one per day.
- Evidence suggests that 44% of users do not customize the HID timeout parameter. This may result in those users having higher irritation rates than otherwise possible.
- User idle periods follow a power law distribution with little temporal correlation, suggesting that a simple predictor could be used to estimate idle period length.
- A combined policy using both HID timeout and presence detection could reduce display energy consumption by 10% compared to the HID timeout policy alone.
- The performance of sonar is quite sensitive to the available audio hardware and operating conditions. 40% of machines in our study were capable of generating and recording significant levels of ultrasound.

2. RELATED WORK

We are aware of no prior work that evaluates DPM policies in practice. The most relevant work studies user interaction with computer systems. The seminal work in that fold is that of Muttka and Livny [12], who studied system idle times, finding that desktop workstations, even in 1990, were idle much of the time. More recent work [4, 1] confirms that this remains the case today. Unfortunately, the traces collected from such studies cannot be used to analyze DPM policies since they are too coarse grained, and/or do not capture HID events.

There appears to be considerable opportunity for DPM policies to reduce overall system power consumption. Mahesari and Vardhan's 2005 study found that the display drew 38% of full system power on an idle laptop computer [10]. Roberson et al. studied the energy consumption of computer displays for the Energy Star program in a set of extensive office surveys [15, 16]. Beyond powering off the display, which has for many years been one of the primary means of reducing energy consumption on deployed machines, Ranganathan et al. describe new display hardware with highly-adjustable reduced power consumption modes [14].

There have been proposals for new DPM policies based on user attentiveness. Dalton and Ellis proposed using a webcam to detect attention for DPM, but their evaluation was incomplete [3]. Visual attention and presence detection techniques are certainly promising; however, further work is needed to determine both their real-world effectiveness and their sensing and processing overheads. The user presence detection policy we evaluate in this paper builds upon a previous study in which we showed that user presence can be detected using computer audio hardware under controlled laboratory conditions [19]. We have now demonstrated an application of the same technique and evaluated it in real-world environments. Our work takes place in the context of the Empathic Systems Project [5], which has generally found that there is a high degree of variation in user satisfaction with any given operating point in a system, and that variation can be exploited by autonomic control that tailors for the individual user. Of relevance to this paper, we have demonstrated a range of user-driven systems for power management [17, 18, 9, 11]. However, none of these studies analyze display power management.

3. DISPLAY ASSUMPTIONS

DPM policies are applicable to both laptop and desktop computers. The obvious motivation for DPM on laptops is to lengthen battery life. Desktop computers have larger displays than laptop computers and therefore their displays consume more energy. Thus, DPM on desktop computers is important from environmental and economic perspectives.

The most recent measurements of display power we are aware of were published in 2005 [10]. To confirm and expand upon those measurements, we measured the total system power of several machines using a clamp ammeter on the AC supply line. By measuring the total electrical current with the LCD display on and off, we can readily calculate the current (and thus power) demand of display. We considered one desktop and three laptops, all running MS Windows.

Figure 1 shows the machines' total system power in several LCD and CPU load states. In the CPU busy state, the OS-reported CPU utilization was ∼100%. We also considered a CPU idle state, which is the common case, as reported in the literature we described in Section 2. Three LCD states were considered: fully off, dim, and full brightness. As we can see from the figure, the display's contribution to system power ranges from 7% to 34%, supporting earlier measurements. Note further that in a very common case, the CPU being idle and the display being bright, the display consumes 31% of system power, and this does not vary much across the machines we tested.

In the remainder of the paper, we report reductions in display energy consumption and system energy consumption as percentages. For the display, a reduction in energy is equivalent to the change in duration that the display is powered on. For a reduction in system energy consumption, we normalize by the 31% figure, making the assumptions (again, supported by previous studies) that the idle state is most common and that that the display is at full brightness. Two examples where these assumptions approximately hold are web browsing and basic office applications. Note that there is no significant energy cost or delay involved in powering up or down the display. We ignore any wear-out costs associated with increased display power switching.

4. DPM POLICIES

We considered two DPM policies in our study. The most important aspect of a DPM policy is how it estimates user attention. We summarize how our two policies do so in

description	CPU type	LCD size	CPU idle			CPU busy
			LCD off	LCD dim	LCD max	LCD max
desktop examples:						
Dell Vostro 410	Intel E8400 3 GHz	22"	63 W	92 (34%)	92 (34%)	128 (25%)
laptop examples:						
Lenovo T61	Intel T7500 2.2 GHz	14.1"	13	14 (7%)	19 (32%)	29 (21%)
Dell Inspiron 8600	Intel Pentium–M 1.6 GHz	15.4"	18	22 (18%)	26 (31%)	37 (22%)
IBM Thinkpad 240	Intel Celeron 300 MHz	10.5"	7	8 (12%)	10 (30%)	16 (19%)
average contribution of LCD to total laptop power:				**12%**	**31%**	**21%**

Figure 1: Total system power consumption in Watts for various system states. Fraction of power due to the LCD is reported in parentheses. Note that desktop LCD backlights typically cannot be dimmed.

policy	attention estim.	failure modes
optimal	true attention	none
HID timeout	input activity	user reading without giving input
presence detection	physical presence as measured by sonar	user present but distracted, sonar errors in detecting presence

Figure 2: Attention estimates for each DPM policy.

Figure 2. A comparison of the two policies can be considered a comparison of two attention estimation models.

4.1 HID timeout

By far the most commonly used policy today is *human interface device (HID) timeout*. HID timeout is familiar to most computer users; it powers down the display after some prescribed time interval has elapsed since the last HID (keyboard or mouse) event. HID timeout has a model of user attention with one parameter: the timeout length T. Under this model, the user is deemed attentive if the time since the last HID event is less than T and, conversely, inattentive if greater than T. HID input resets the interval and powers up the display.

This model works in some cases because it captures engagement between the user and computer. However, attention is not well measured: input activity-based techniques are unable to distinguish between a truly inattentive user and one who is actively reading the display without using the mouse or keyboard. Thus, the model parameter T is typically set to a conservative value: at least five minutes. Microsoft Windows' default value is five minutes for battery-powered operation.

4.2 User presence detection

The proposed user presence detection policy attempts to use sensor-based measurement of user presence as a user attention estimate. In earlier work, we showed how sonar can be used to detect user presence on a standard laptop computer without any additional hardware [19]. The laptop computer emits a sound "ping" from its speaker and records the resulting echo on its microphone. The ping is a 22 kHz sine wave. This frequency is chosen to be high enough to be silent to humans (thus it is *ultrasonic*), yet low enough that it might be produced and recorded by the audio hardware built into most laptop computers. To sense the user, the computer can determine the amount of variation in the ping's echo. Little variation indicates a still room; more variation indicates motion in the environment, which may indicate the presence of a human. The technique is very effective under controlled laboratory conditions.

We use a threshold to determine whether a sonar reading indicates user presence or absence. High sonar reading variation corresponds to presence and low variation to absence, but calibration is required to determine a threshold value. In our lab experiments [19], sonar readings were taken while the user was known to be present and also while absent and a threshold value between these two readings was chosen. This calibration procedure works fine if the acoustic environment is static. However, if the computer is moved between different environments or if the environment's acoustics change then calibration must be repeated.

Calibration is one of several practical issues that arise in real-world sonar presence sensing; we have addressed these as follows. It is easy to record sonar readings representative of a present user: for example, readings can be taken while the mouse is is motion. In order to avoid requiring the user to leave the computer to calibrate an absence reading level, we approximate absent sonar readings as zero and simply set the threshold to one half of the present value. Sonar adapts to changes in the speaker volume levels by normalizing the reading to the recording's ping intensity. Sonar uses only a narrow ultrasonic frequency band and it does not prevent normal use of the speakers, for example in playing music. In informal laboratory tests, playing music had no noticeable effect on sonar performance. If an irritation event occurs, this indicates that presence detection failed. To prevent repeated irritation, a factor of 0.8 is applied to lower the threshold after each irritation event. Only the next hourly recalibration may increase the threshold.

5. IMPLEMENTATION

We employ both the HID timeout and user presence detection policies within a single piece of software, the Sonar Power Manager, a utility that we have designed for use with Windows XP, Vista, and 7, as well as Linux. Sonar Power Manager is designed to simultaneously test both policies by recording the actions of both policies in parallel.

If the sonar presence sensor indicates that the user is absent the display is shut off. Similarly, if the input activity timer expires the display is shut off. The timeout value is set equal to the display sleep timeout in the Windows control panel for battery-powered operation. Thus, the presence detection policy adds an additional display shut-off criterion

to the existing Windows HID timeout policy. By design, it is more aggressive than the default policy.

The Sonar Power Manager is a user-space utility implemented in C++ with the WxWidgets and PortAudio libraries to achieve portability across Windows and Linux. The software is open source and has been released with a permissive license. Both the source code and a Windows executable can be downloaded from our web site[1]. The website also describes the software's operation in more detail.

The Sonar Power Manager uses minimal power and CPU time. On a Lenovo T61 laptop, sonar sensing consumes ~3% of the available CPU cycles on one core. The measured system power overhead due to sensing for the machines listed in Figure 1 was between 4% and 10%.

6. USER STUDY

We conducted a user study to collect usage traces appropriate for evaluating DPM.

Recruitment and users. To assist recruiting participants, a Northwestern University press release on our work led to a posting on the popular computer and technology news website `slashdot.org`. Shortly thereafter, the Sonar Power Manager was downloaded over 10,000 times. It is from these downloaders that we draw our data. Hence, it is important to note that our user sample represents those with an interest in technology news and may therefore may react differently to the HID timeout and user presence detection policies than would the general population.

Sonar Power Manager collects log data on the operation of the policies it is testing. Users who installed the utility were given the option of "opting-out" of sending these logs to us. The majority of users opted-out of logging. It is also important to note that users who did opt-in ran the software for varying amounts of time. A user could also opt-out of logging at any time, and the logging process was automatically disabled after one week. We eliminated logs from those who used the software for less than one hour.

Log contents. For those who opted-in, Sonar Power Manager collected logs of sonar and power management events. Logs were compressed and sent to our servers every hour. The following information was time stamped and logged:

- The starting and ending time of HID input activity.
- Irritation events for both policies. Specifically, an irritation event is defined as a HID event that causes the display to wake less than 5 seconds after it was slept.
- The value of each sonar measurement.
- Times at which logging starts and stops.

Recall that Sonar Power Manager collects log data for both the HID timeout policy and the user presence detection policy simultaneously

High-level view of the dataset. We acquired 3,738 hours of usage by 181 volunteers, each with a different machine. There were 177 Windows users and 4 Linux users. Users ran the software for varying amounts of time. Figure 3 gives the distributions of the interval of time that the software was installed and the amount of time the software ran with full
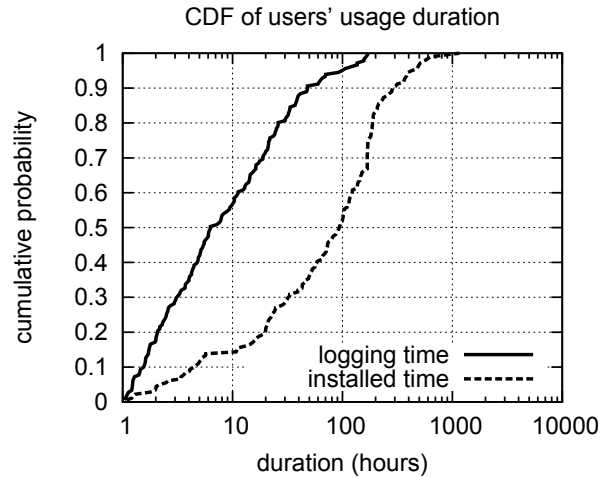

Figure 3: CDF of installed time and logging time. The difference is the time that the computer was off or logging was disabled.

logging. The difference between these two durations is the time the machine was off or the user had logging disabled. The median amount of full-logging time was seven hours.

7. RESULTS

We analyzed the logs collected during the user study to make the observations given in Section 1. We present our results in several parts: a characterization of user active and idle periods, an analysis of the effectiveness of the HID timeout policy, an analysis of the upper bound of energy savings for DPM policies based on user behavior, an analysis of the effectiveness and tradeoffs of the user presence detection policy, and a characterization of sonar in practice.

In our treatment, we generally refer to two kinds of results: *aggregate* results are computed after first joining data collected from all users, while *individual* results are first computed for each user and then afterwards averaged across users. Aggregate results weigh each observation equally while individual results weigh each user equally. Since users provided different amounts of log data, these two averaging methods give different results. We also use logarithmic plots because it is well known that human perception of time durations scales logarithmically [6, 8, 13].

7.1 Characterization of idle/active periods

Computer usage may be broken into a sequence of alternating active and idle periods, which we identify in our traces based on HID events. We consider a HID-active period as one in which HID events are spaced by no more than than one second. HID-active periods are separated by HID-idle periods which we define as those where more than one second elapses with no HID events.

Figure 4 shows the distribution of idle and active periods as defined above that were found in our data. Notice that the idle period distribution clearly follows a straight line in this log-log plot, indicating that it is a power-law distribution. Active periods tend to be shorter than idle periods and they seem to follow a piecewise power-law distribution (note
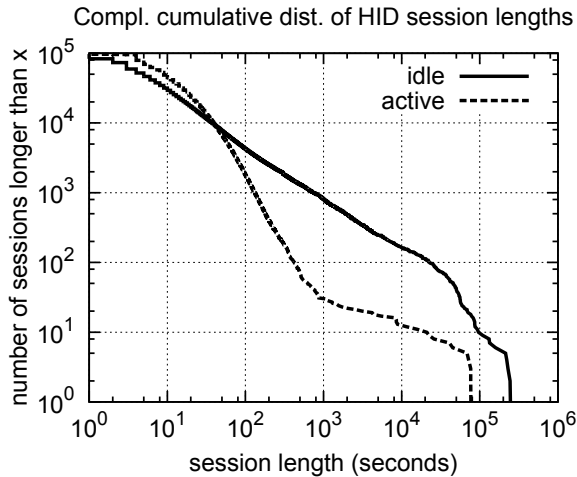
Figure 4: Distribution of lengths of HID-active and HID-idle periods.
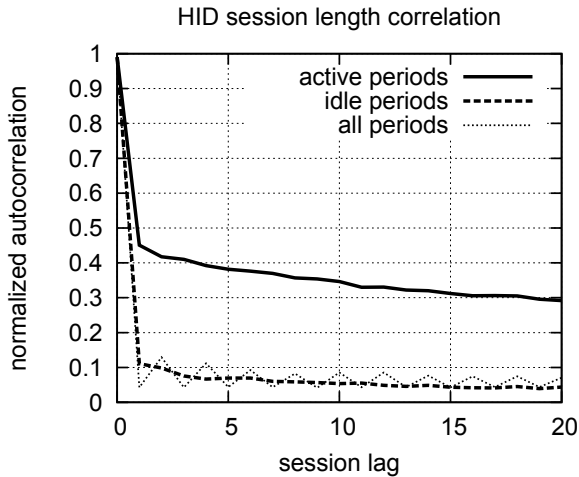

Figure 5: Autocorrelation of consecutive active and idle periods. These are individual results.

that the power-law exponent changes around 1000 seconds, evidenced by a change in slope).

In Figure 5 we show an autocorrelation analysis of active and idle sessions, considering active→active, idle→idle and (idle+active)→(idle+active). Although there is significant correlation of active periods to themselves, there is little to no correlation that might support the prediction of idle periods. This suggests that to predict an idle period's length, the best approach is probably simply to exploit the memory property of the idle periods' power-law distribution [7].

7.2 HID timeout policy performance

Timeout setting distribution. Figure 6 shows the distribution of users' DPM timeout settings. Given the observations we have repeatedly made in the Empathic Systems Project (see Section 2), we might expect that users vary consider-
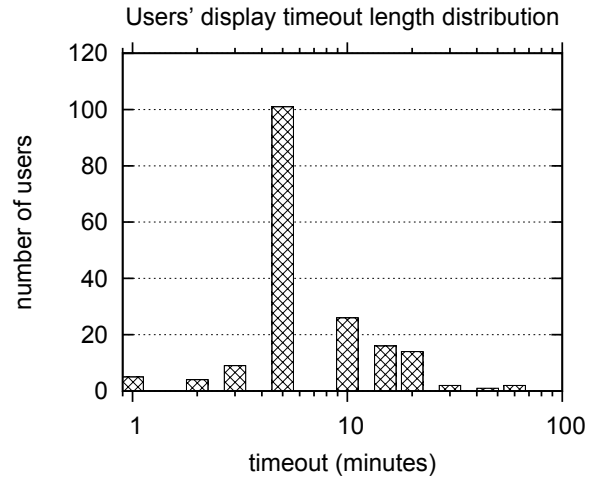

Figure 6: Distribution of users' HID timeout values.

ably in their performance expectations and that this should be reflected in a wide range of DPM timeout settings. This is indeed the case.

The peak of users at 5 minutes appears to be due to this being the default value in Windows. In Section 7.3 we will explore the implications of this peak.

Energy savings. Figure 7(a) shows the distribution of the energy savings achieved by the HID timeout policy for users in our study. The savings is measured by the fraction of total runtime during which the display was shut off–called the *sleep fraction.* The median sleep fraction was 25%.

Figure 7(b) shows the sleep fraction as a function of users' timeout values. One might expect that the sleep fraction would decrease monotonically with the timeout value, but this is not the case. The mean aggregate sleep fraction across all users is about 51%. Although there there is some trend of lower energy savings with increasing timeout value, clearly the variation among users swamps this trend. Multiplying the display sleep fraction by the fraction of system power due to the display (which we approximate in Section 3 as 31%) gives the total system energy savings. The average total system power savings seen in our study is thus 51%×31%=15.8%.

It is important to note that the data of Figure 7(b) are based on users choosing their own timeout values. That is, it is an observational result, not intervention-based result. If we were to force different timeout values in a controlled way, the curve might be quite different.

Irritation rates. User irritation occurs when the policy shuts off the display while the user is actually attentive. Of course, it is highly likely that the user will respond to such an *irritation event* with some HID input to wake the display. The sequence of the display being powered down quickly followed by HID input thus labels irritation events. In the following, we consider any display sleep period shorter than five seconds to represent an irritation event.

Figure 8 shows the user irritation rate as a function of the user's timeout setting. Note that for all timeout settings,
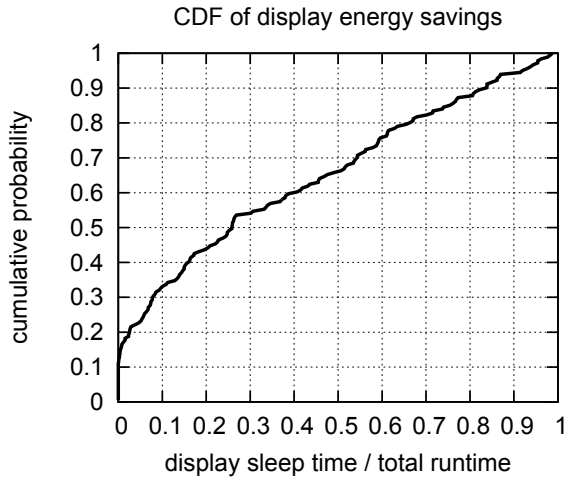
(a) CDF of display energy savings achieved by the HID timeout policy.
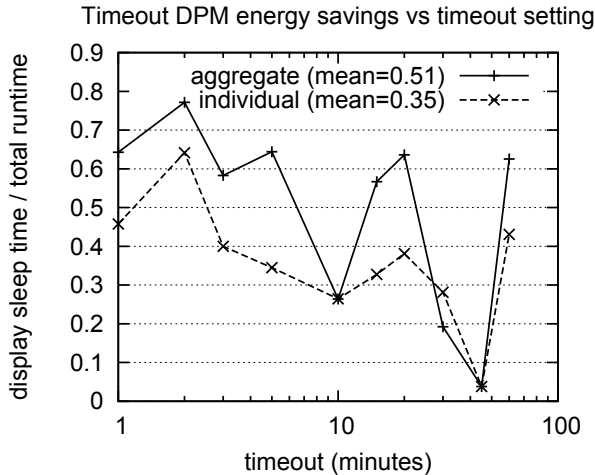
(b) Display energy savings achieved as a function of the user's HID timeout parameter.

Figure 7: Users' achieved display energy savings.

except the lowest setting (one minute), irritation rates are quite low, on the order of once per day. We discuss the surprising peak in the curve at 5 minutes next.

7.3 Default users

Surprisingly, Figure 8 shows that users who chose a timeout value of two or three minutes experienced less irritation than those with a higher timeout value of five minutes. Five minutes happens to be the Windows default value, and it is used by roughly half of the users (see Figure 6). We suspect that the peak in irritation at five minutes is due to a significant group of users who neglected to customize the timeout value, and thus have more irritation and/or lower energy savings than would otherwise be possible.

Some portion of the five minute users have deliberately chosen five minutes, while the rest use it by default. We now attempt to estimate the sizes of these groups. We first ignore the problematic five minute users while fitting a distribution curve to the remaining users. Then we can use the distribution to estimate the expected size of the deliberative
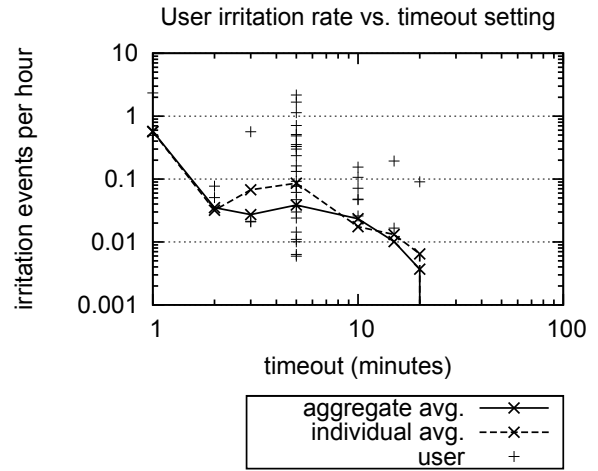

Figure 8: User irritation rate as a function of the user's DPM timeout setting.
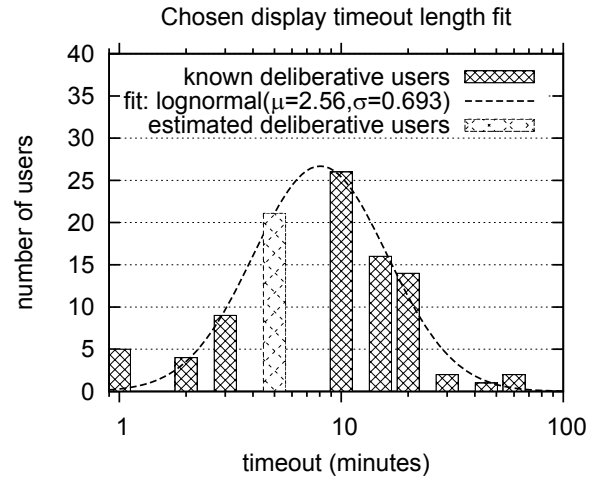

Figure 9: Distribution of deliberative users from Figure 6. The curve is fit to all users except those with the default five minute value.

group. Figure 9 shows that a log-normal distribution appears to fit user timeout preference well, with PDF defined as follows:

$$PDF_{lognormal}(x) = \frac{1}{x\sigma\sqrt{2\pi}} \exp\left[-\frac{(\ln x - \mu)^2}{2\sigma^2}\right]$$

Using this fit, we estimate that, of the 101 users with the default five-minute timeout, 21% deliberately chose that value while 79% just use it by default. This latter group is significant in that it represents 44% of the users in our study. Due to the nature of our participants, it is likely that this latter figure significantly underestimates the proportion of users in the general population who ignore the customization controls for HID timeout policies. Assuming that the peak in irritation seen in Figure 8 is due to this population, irritation rates can be reduced by somehow forcing users to optimize the timeout setting.
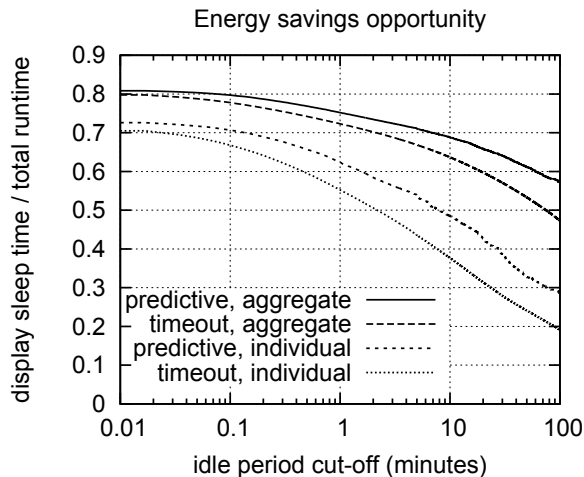
Figure 10: Energy savings opportunity as a function of the user attention cutoff. Predictive and timeout policies are compared, as described in the text.

7.4 Energy savings upper bound

Figure 10 shows the upper bound for display energy savings that any DPM policy can achieve. Here we assume that the display can be powered-down whenever there is no HID input for a given duration; we call this duration the idle period cut-off. We include all idle periods encountered in our data, not just those where the user was inattentive. Because of this, the upper bound is loose–a policy that captured actual user attention would have lower energy savings. At most, the display energy savings are 81%, for a system energy savings of 81%×31%=25%.

Two classes of policies are compared. The predictive policy assumes foreknowledge of an idle period's length and thus allows the display to be powered down for the entire duration of the period. The timeout policy powers down the display only after an idle period extends beyond the cut-off value. The closeness of these upper bounds suggests that using a sophisticated idle period length prediction scheme will give little benefit. Furthermore, the diminishing returns of targeting brief idle periods is seen as the curves flatten out below about one minute. Conversely, the steeper slopes around the default timeout value of five minutes indicate that energy savings are sensitive to the timeout setting in this regime.
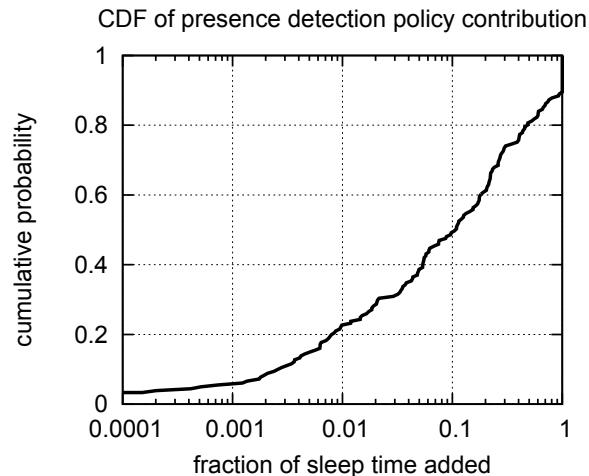
7.5 User presence detection policy results

We now show the benefit to DPM of adding a sonar presence detection sensor. Initially, we consider all users, but later we attempt to isolate the machines on which sonar presence sensing worked well.

Energy savings. Figure 11(a) shows a comparison of the sleep times (and thus the energy savings) of the HID timeout and sonar presence detection policies when they were run simultaneously. Many users (those above the diagonal) experienced more energy savings due to the HID timeout policy than presence detection alone. For the users below the diagonal, presence detection alone performed better than HID timeout. We begin by considering a simple combined
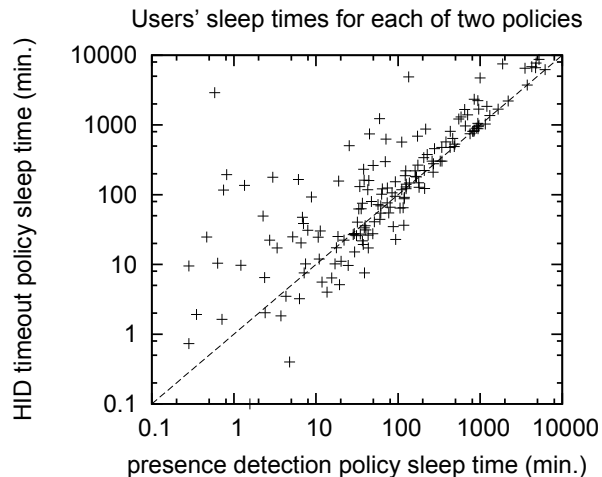
(a) Users below the diagonal line had more energy savings due to the presence detection policy than the HID timeout policy.

(b) CDF of display sleep time in the combined presence-timeout policy which was due exclusively to the presence detection policy.

Figure 11: Comparison of energy savings due to presence detection and HID timeout policies.

policy in which both HID timeout and presence detection policies are run in parallel. Either policy can decide to turn off the display, and HID input turns it back on.

Figure 11(b) considers the additional energy savings that sonar provides in the combined policy. It shows the energy savings that would be lost if sonar presence sensing were disabled. In particular, the CDF shows that for half of the users, presence detection contributes at least 10% of the total display energy savings. Also, the presence detection policy at least doubled display energy savings for almost 20% of users.

Sensing overhead. There is a small energy overhead associated with sonar measurement so we must judge whether the gains from incorporating sonar outweigh the costs. Figure 12 shows the ratio of the gains to the number of sonar measurements. We extrapolate the energy overhead based
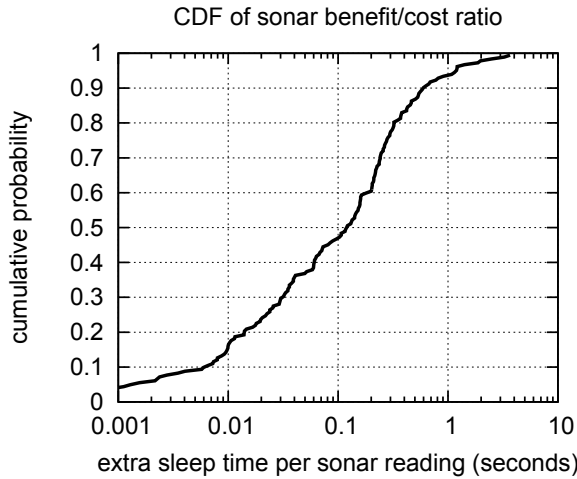
CDF of sonar benefit/cost ratio

Figure 12: CDF of additional display sleep time per ∼1 s sonar reading.
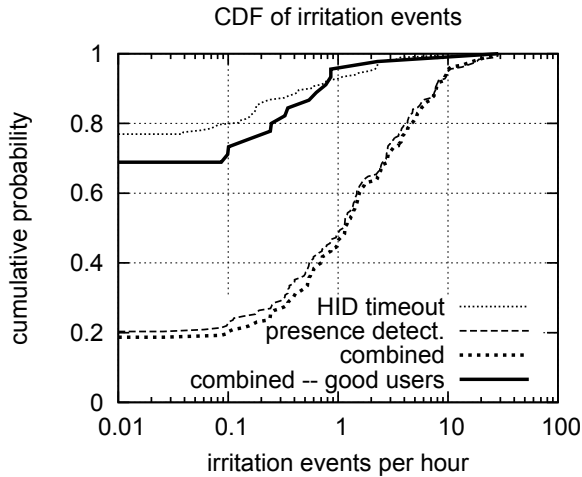
CDF of irritation events

Figure 13: CDF of irritation events per hour.

on the 7% average sensing power overhead recorded on lab computers (see Section 5). In Figure 12, the top 10% of users gained at least 0.6 s of display sleep time per 1 s sonar reading. Costs are about $(7\% \times 1\,\mathrm{s})/(31\% \times 0.6\,\mathrm{s}) = 38\%$ of gains for these users. For users with less than 0.22 s of extra display sleep time per sonar reading, costs were at least at large as gains. Sonar can and should be disabled for these 67% of users.

Irritation rates. While Figure 8 showed low user irritation rates for the HID timeout policy, Figure 13 shows more significant irritation rates for the presence detection policy. The presence detection policy's median value of one irritation event per hour is relatively low but significantly higher than the HID timeout policy's rate of one per day. The combined policy's irritation rates track those of the pure presence detection policy. The curve annotated "combined – good users" shows the performance of the combined policy for situations where it is most effective, as discussed next.

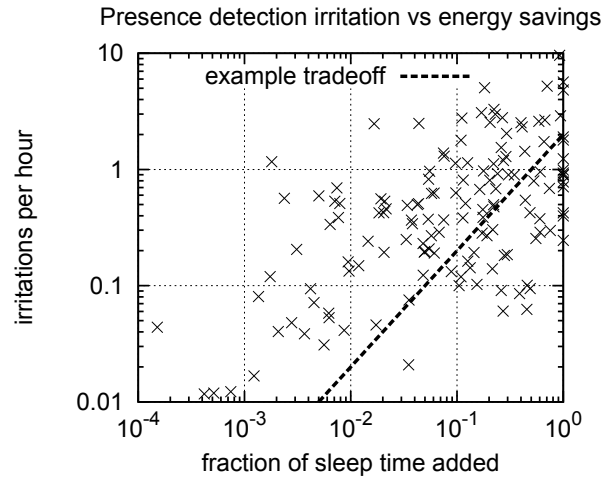## Presence detection irritation vs energy savings

Figure 14: Relationship between irritation rates and energy savings for the presence detection policy. The 44 users below the dashed line experienced fewer than 2 irritation events per hour of sleep due to presence detection.

Figure 14 shows that there is some correlation between energy savings and irritation rates for the combined policy. As expected, at the left side of the plot, we see that users for whom sonar rarely powered off the display were rarely irritated. Note that some users with high energy savings rates did not experience high irritation rates (users below the dashed line in Figure 14). These are the users for whom sonar provided a real benefit over the pure HID timeout policy.

Thus far, we have explained a combined HID timeout and presence detection policy that simply runs the two in parallel and allows either to turn off the display. A more advanced policy would determine when presence detection is likely to be effective. One approach would be simply to run the two policies to collect information similar to that of Figure 14. The user would then choose a dividing line (as shown in the figure) to trade off between irritation and energy savings. The example line represents a tradeoff that requires fewer than 2 irritation events per hour of additional display sleep time. The "combined – good users" curve in Figure 13 reflects the effects of the example line. Another combined policy might track the irritation rates of the two policies and choose the one that currently has the lowest rate, over some window. This would be an application of the multiple experts strategy [2].

7.6 Characterizing sonar

Our study also allowed us to characterize the performance of ultrasonic sonar on ordinary computers in ordinary situations. Our previous analysis [19] focused on a small range of audio hardware under controlled lab conditions.

Sonar-based detection of user presence requires audio hardware that is capable of both emitting and recording ultrasound. Sonar Power Manager was configured to conduct a calibration phase and report its findings back to us. Because calibration happens at startup time, we have calibration data from an order of magnitude more users and machines
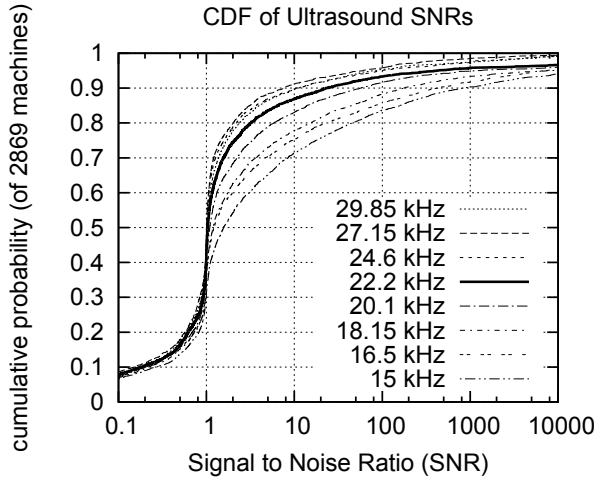
(a) SNR spectrum calculated from white noise stimulus.
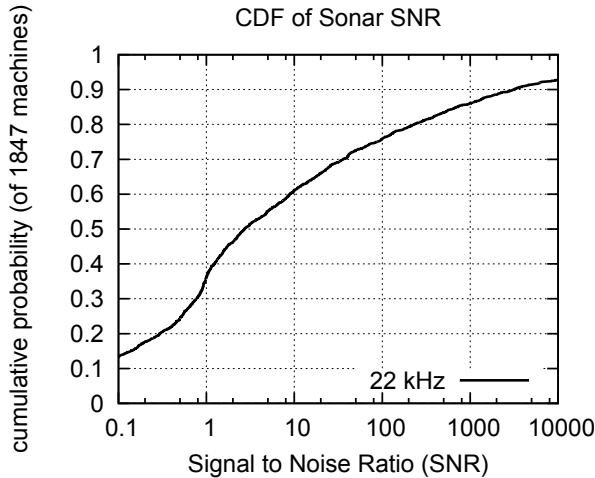
(b) SNR calculated from a single sine wave.

Figure 15: Ultrasound signal to noise ratios (SNR). On a given machine, a low SNR means that sonar will not work.

than we have long-term logging data. The important part of the calibration data is a measurement of the ultrasound signal to noise ratio (SNR). This is accomplished by generating and simultaneously recording ten seconds of white noise in the range 15 to 30 kHz, as well as recording ten seconds of silence. The white noise is the desired signal and the silence is the background noise. Their power ratios at different frequencies give an approximate SNR spectrum, for which we show the CDF for 2,869 machines in Figure 15(a). Figure 15(b) shows the SNR calculated from recordings of a 22 kHz sine wave. This measurement occurred after the previous one, and so there is an attrition of users—only 1847 machines are included. This measurement method reduces the effect of spectral leakage and interference, so it is more reliable.

As a rule of thumb, we do not expect sonar to work if the SNR is below ten. As expected, lower frequencies give a generally higher SNR at the expense of becoming audible to more humans. Note that at the frequency used by the experimental software (22 kHz), about 40% of machines had a
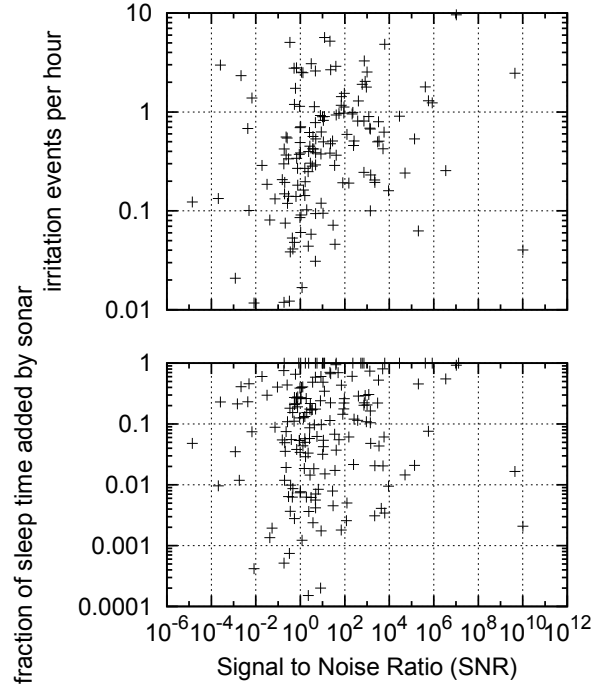

Figure 16: Correlation between SNR and sonar-induced irritation (first and second plots) and energy savings (third plot).

good SNR. We might expect users with high irritation rates seen in Figure 13 and low cost-benefit ratios in Figure 12 to fall into the 60% of machines with low sonar SNR. However, this did not appear to be the case. Figure 16 shows correlation between SNR and both irritation and energy savings. Strangely, a clear correlation is not evident. Based on these results, it seems clear that any combined sonar and HID timeout policy should select between the two using the performance of the overall system, rather than the SNR found during calibration. Ideally, an ultrasound calibration technique capable of predicting sonar presence detection performance would be found.

As previously stated, Figure 15(b) tells us that around 40% of available machines were capable of generating and recording ultrasound with their current audio systems. While it straightforward to understand why a commodity microphone can capture ultrasound, it is surprising that speakers are capable of generating it. We attribute this to the lack of low pass filtering in common output paths combined with cheap speaker elements being quite small.

8. CONCLUSIONS AND FUTURE WORK

We have presented the first study of the real-world behavior of display power management (DPM) policies, considering the widely-used HID timeout policy, a sonar-based presence detection policy, and a combined policy. We found that user idle periods follow power law distributions with little temporal correlation. The maximum possible reduction in energy used for the display is 81%, while the HID timeout policy manages to achieve 51%. Our results suggest that 56% of users have already customized the HID timeout

policy on their machines resulting in surprisingly low levels of user irritation of around once per day. However, 44% of users have not, and these users experience more irritation. The presence detection policy combined with the HID timeout policy further reduced display energy consumption by 10%. 40% of available machines can generate and record ultrasound effectively.

Based on the experimental results, we recommend modifying Operating System DPM implementations as follows. We have described a method of detecting user irritation events caused by false display shutoff policy decisions. Whenever this happens, the user should be prompted to optionally lengthen the DPM timeout. This feedback scheme would allow users to easily fix a sub-optimal policy setting which might otherwise be ignored. Alternatively, a fully autonomic policy might increase the timeout length by 10% whenever an irritation event occurs while reducing the timeout by 10% each active hour. We also recommend adaptively adding sonar presence detection to the DPM policy. During runtime, if either the ultrasound SNR or achieved display energy savings are too low or the irritation rate is too high, sonar should be disabled.

So far we have considered only DPM policies for current hardware. Several additional paths exist. First, better user attention detection sensors could be employed, for example the biometric sensors we have begun to use to gauge user satisfaction [18]. Dedicated presence sensors (e.g., infrared motion sensors) can be both reliable, energy efficient, and inexpensive. We also note that most current laptop computers do not even have a button or switch to turn off the display. Another research direction is DPM policies for emerging display technologies. In particular, e-paper displays consume power only when the display contents change. LED backlit displays (which are already replacing traditional fluorescent backlit displays) have lower overall power consumption. Some experimental displays have been demonstrated which allow only part of the display to be lit. The proliferation of new computer displays throughout homes and workplaces might provide an additional motivation for DPM in reducing visual clutter/confusion as well as saving power.

9. REFERENCES

[1] BEAUVISAGE, T. Computer usage in daily life. In *Proc. Intl. Conf. on Human Factors in Computing Systems* (Apr. 2009), pp. 575–584.

[2] BLUM, A., AND BURCH, C. On-line learning and the metrical task system problem. In *Proc. Conf. on Computational Learning Theory* (July 1997), pp. 45–53.

[3] DALTON, A. B., AND ELLIS, C. S. Sensing user intention and context for energy management. In *Proc. Wkshp. on Hot Topics in Operating Systems* (May 2003).

[4] DINDA, P. A. The statistical properties of host load. *Scientific Programming 7*, 3–4 (1999), 211–229.

[5] DINDA, P. A., MEMIK, G., DICK, R. P., LIN, B., MALLIK, A., GUPTA, A., AND ROSSOFF, S. The user in experimental computer systems research. In *Proc. Wkshp. on Experimental Computer Science* (2007).

[6] EMBLEY, D. W., AND NAGY, G. Behavioral aspects of text editors. *ACM Computing Surveys 13*, 1 (Jan. 1981), 33–70.

[7] HARCHOL-BALTER, M., AND DOWNEY, A. B. Exploiting process lifetime distributions for dynamic load balancing. In *Proc. Intl. Conf. on Measurement and Modeling of Computer Systems* (May 1996), pp. 13–24.

[8] KOMATSUBARA, A. Psychological upper and lower limits of system response time and user's preference on skill level. In *Proc. Intl. Conf. on Human Factors in Computing Systems* (Aug. 1997), pp. 829–832.

[9] LIN, B., MALLIK, A., DINDA, P., MEMIK, G., AND DICK, R. User- and process-driven dynamic voltage and frequency scaling. In *Proc. Intl. Symposium on Performance Analysis of Systems and Software* (Apr. 2009).

[10] MAHESRI, A., AND VARDHAN, V. Power consumption breakdown on a modern laptop. *Power-Aware Computer Systems 3471* (2005), 165–180.

[11] MALLIK, A., COSGROVE, J., DICK, R. P., MEMIK, G., AND DINDA, P. Picsel: measuring user-perceived performance to control dynamic frequency scaling. In *Proc. Intl. Conf. Architectural Support for Programming Languages and Operating Systems* (2008), pp. 70–79.

[12] MUTKA, M. W., AND LIVNY, M. The available capacity of a privately owned workstation environment. *Performance Evaluation 12*, 4 (July 1991), 269–284.

[13] NEWMAN, W., AND LAMMING, M. *Interactive System Design*. Addison-Wesley, Boston, MA, USA, 1995.

[14] RANGANATHAN, P., GEELHOED, E., MANAHAN, M., AND NICHOLAS, K. Energy-aware user interfaces and energy-adaptive displays. *Computer 39*, 3 (2006), 31–38.

[15] ROBERSON, J. A., HOMAN, G. K., MAHAJAN, A., NORDMAN, B., BROWN, C. A. W. R. E., McWHINNEY, M. C., AND KOOMEY, J. G. Energy use and power levels in new monitors and personal computers. Tech. rep., Lawrence Berkeley National Lab, Berkeley CA., July 2002.

[16] ROBERSON, J. A., WEBBER, C. A., McWHINNEY, M. C., BROWN, R. E., PINCKARD, M. J., AND BUSCH, J. F. After-hours power status of office equipment and inventory of miscellaneous plug-load equipment. Tech. rep., Lawrence Berkeley National Lab, Berkeley CA., Jan. 2004.

[17] SHYE, A., OZISIKYILMAZ, B., MALLIK, A., MEMIK, G., DINDA, P., AND DICK, R. Learning and leveraging the relationship between architectural-level measurements and individual user satisfaction. In *Proc. Intl. Symposium on Computer Architecture* (June 2008).

[18] SHYE, A., PAN, Y., SCHOLBROCK, B., MILLER, J. S., MEMIK, G., DINDA, P., AND DICK, R. Power to the people: Leveraging human physiological traits to control microprocessor frequency. In *Proc. Intl. Symposium on Microarchitecture* (Nov. 2008).

[19] TARZIA, S. P., DICK, R. P., DINDA, P. A., AND MEMIK, G. Sonar-based measurement of user presence and attention. In *Proc. Intl. Conf. on Ubiquitous Systems* (Sept. 2009), pp. 89–92.