How Much is Performance Worth to Users?

Adam Hastings
Columbia University
New York, NY, USA
hastings@cs.columbia.edu

Lydia Chilton
Columbia University
New York, NY, USA
chilton@cs.columbia.edu

Simha Sethumadhavan
Columbia University
New York, NY, USA
simha@columbia.edu

ABSTRACT

In computer systems design, computer architects can evaluate features and techniques in terms of traditional design metrics like power, performance, and area but not in terms of net benefit or cost to the end user. For example, a security feature may come at a 10% cost to performance, but is this worth the tradeoff? The problem is that user-level features like security, privacy, and usability can be converted into a monetary amount but traditional architecture metrics (which lack the notion of user value) cannot. In this paper, we make the first known attempt to bridge this gap: We conduct two studies (one of which is incentive compatible) that elicit the value of performance in terms of US$ to end users. Thus in this work, we make the first known quantitative measurement of the tradeoff between performance and user value, providing architects with a novel design metric and filling a crucial gap in the end-to-end quantitative evaluation of systems.

CCS CONCEPTS

• **Security and privacy → Economics of security and privacy**; *Security in hardware*; *Systems security*; • **Human-centered computing → Empirical studies in HCI**.

KEYWORDS

user studies, performance measurements, human factors in computing, security economics, security tradeoffs

ACM Reference Format:
Adam Hastings, Lydia Chilton, and Simha Sethumadhavan. 2023. How Much is Performance Worth to Users?. In *20th ACM International Conference on Computing Frontiers (CF '23), May 9–11, 2023, Bologna, Italy*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3587135.3592194

1 INTRODUCTION

A quantitative approach to computer architecture and systems design has been a key factor in achieving meaningful improvements for several decades. Computer architects and designers have been tremendously successful at developing metrics to measure important quantities such as performance, power consumption, die area, and reliability, which have allowed systems designers to have a clear conversation about pros and cons of competing approaches. More recently, architects and designers are increasingly tasked with designing systems for user-facing requirements like responsiveness, security, and privacy, which can be broadly classified as a quality requirements [2, 8, 12, 27]. However, unlike traditional metrics, these user-facing requirements are often not easily measurable, making it difficult for architects to quantitatively determine which tradeoffs are worth making.

For example, should a phone designer add biometric authentication if it comes at the expense of storage space? How much device responsiveness should be exchanged for an always-on security feature? These types of questions are typically unanswerable using traditional design metrics (like power or performance) because traditional design metrics are agnostic to how much various design features are worth to users. Yet deciding these tradeoffs is a necessary and consequential step of the design process. How can systems designers and computer architects more rigorously balance design requirements like power, performance, and die area against indeterminate requirements like user preferences?

In this work, we make the first known attempt to introduce user preferences as a quantifiable metric for design decision-making. Specifically, we aim to put a price to users' value of performance. In other words, this work finds the "exchange rate" between performance and user value, in terms of US$. By establishing this "exchange rate", we provide systems designers and architects with a quantitative metric by which they can balance tradeoffs between performance and other features which may "cost" performance (such as security or usability features, among others). To illustrate, suppose a systems architect must decide whether or not to include an image processing accelerator in a system that, if included, would come at the expense of cache sizes and decrease general system performance by 10%. Is this worth the cost? Via our methodology, an architect can put a monetary amount on the opportunity cost (in terms of user satisfaction) of such a feature, and can compare this to users' value of the feature itself (perhaps derived via market research studies). Our work makes this type of quantified decision-making possible.

To find the worth of performance to users, this paper conducts two complementary experiments. The first experiment asks participants to run a program on their personal computing device and perform simple everyday tasks while enduring throttled device performance. Using a series of yes/no questions, we then elicit how much each participant would have to be paid to accept a permanent device performance loss of either 10%, 20%, or 30%. For performance losses of 10%, 20%, and 30%, we find this amount to be $381 (N=21), $457 (N=24), and $823 (N=22), respectively.

For the second experiment, we develop an incentive compatible experimental protocol, meaning that participants are incentivized to make decisions and answer questions according to their true preferences. Specifically, participants are given choices between computer performance and real-world money; participants who

CF '23, May 9–11, 2023, Bologna, Italy

ACM ISBN 979-8-4007-0140-5/23/05…$15.00
https://doi.org/10.1145/3587135.3592194

choose the money over performance must then endure throttled device performance on their personal computing devices for up to several days at a time. Instead of finding participants' valuation of a *permanent* device performance loss, this methodology finds the participants' valuation of a *per day* performance loss (which is a necessary consequence of our choice in experimental design). Using participants' responses, we find that users would trade performance losses of 10%, 20%, and 30% to their personal devices in exchange for $2.27 per day (N=26), $4.07 per day (N=29), and $4.44 (N=30) per day, respectively.

In addition to the above two methodologies, we also perform a large-scale online survey of the same questions asked in the previous protocols, albeit with no hands-on throttled device interaction and with no incentive compatibility. We do this to 1) observe a larger sample size, and 2) compare with the other methodologies' results to determine whether or not the hands-on interaction and incentive compatibility is necessary for this type of experimentation. From our large online survey results, we find that participants report that in order to accept a performance loss of 10%, 20%, or 30%, they would need to be offered at least $11.15, $22.85, and $24.92 per day, respectively (N=306), or $499, $1214, $3723 in total, respectively (N=306); these results are much higher than those obtained via experimentation, suggesting that the effort of in situ experimentation is necessary, and that this type of experimentation cannot be replicated via a simple survey mechanism.

The rest of the paper is organized as follows: We provide justification for our choice of experimental design in Section 2, namely why we choose to measure participants' willingness to accept performance losses. We then describe our methodologies in Section 3, and present results in Section 4. This is followed by a discussion on some of the results in Section 5. Related work is reviewed in Section 6. Finally, this paper concludes in Section 7.

2 WHY MEASURE THE COST OF PERFORMANCE *LOSSES*?

In our efforts to quantify the value of performance, we chose to measure participants' willingness to accept (WTA) performance losses on their personal devices[1]. Although perhaps not the most obvious way to study this problem, we find it to be the most appropriate value to measure given the constraints: First and foremost, to get an accurate measurement of user preferences, we felt it was necessary to study users in their own environment and on their own computing devices. This ruled out in-the-lab style experimentation[2]. Given this constraint, the only direction by which we can reliably adjust users' devices' performance is downward (after all, if there was an easy way to permanently increase device performance, many users would have done so already!).

Another reason for measuring users' willingness to accept performance losses is that it allows for systems and architectural features to be evaluated and compared by their opportunity cost in terms of both performance *and* user preference. To illustrate this point, consider a systems architect who wants to include hardware support for secure memory bounds checking. At the systems and architectural level, the opportunity cost of such a feature is the performance gain that could otherwise be achieved without the bounds checking, while at the end user level, the opportunity cost is the additional features that this extra performance could allow. By quantifying the relationship between performance and end user value, we provide systems designers and architects a means by which they can translate from the low-level domain of systems and devices to the high-level domain of user value.

Finally, by measuring the cost of performance losses, we also make it possible measure the in-the-field cost of security patches to hardware vulnerabilities like Meltdown and Spectre (see Section 5 for more details).

2.1 How Do We Throttle Participants' Devices?

Given the above constraints, we needed a method to remotely and reliably throttle the performance of study participants' personal devices. We found that the best way to do this was by throttling CPU frequency. On Windows devices, this is achieved via the `powercfg` command line utility by adjusting the value of `PROCTHROTTLEMAX`, which limits the systems' CPU frequency relative to its maximum (i.e. setting `PROCTHROTTLEMAX` to `50` on a 4.0 GHz device should lower CPU frequency to 2.0 GHz). We limited experimentation to participants using Windows 10 devices[3].

Of course, our experiments' validity hinges on whether or not CPU frequency is a good enough dial by which we can adjust device performance (which is much more than a scalar value and depends on a multitude of systems components and specs, as well as usage). Put another way, it is unclear if an $X\%$ drop in CPU clock speed will produce an $X\%$ loss of performance. To answer this question, we benchmarked two Windows 10 devices—a laptop and a desktop device[4]—with three device performance benchmark suites—SPECspeed 2017 Integer, SPECrate 2017 Integer, and WebXPRT 3—at various clock speeds. Results are show in Figure 1. All three benchmarks suites show a highly linear relationship between CPU frequency and benchmark score: Pearson correlation coefficients are above 0.996 for all datasets except for the SPECrate 2017 Integer benchmark on the i7-8550U laptop (which had a Pearson correlation coefficient of 0.918). Based on these results, we find CPU frequency to be a reasonable proxy for device performance, and find it reasonable to assume that decreasing participants' CPU frequencies by $X\%$ will decrease performance by roughly $X\%$. Given this, and

[1] Willingness to accept (WTA)—a concept borrowed from the field of economics—is the minimum amount that a person would have to be paid to accept some unfavorable condition or outcome.

[2] The justification here is that testing participants in a tightly-controlled laboratory environment would require experimentation on different devices (with different performance specs) than what the participant might be used to; hence it would be inappropriate to measure participants' value of performance against such a contrived setting. For example, questions of "how much would you have to be paid to make this device 10% slower" are meaningless unless participants already have a baseline understanding of how fast or responsive a device is in the first place. Hence we chose to study users only in their own environment and on their own personal devices.

[3] Our reasons for this were that 1) Windows 10 comprises a majority of desktop and laptop users, and 2) we could not find a reliable method for throttling CPU frequency on MacOS devices. On Linux devices, throttling CPU frequency is easy but the share of desktop and laptop Linux users remains small. We also considered experimenting on mobile phones, but were not able to throttle device performance without asking study participants to "jailbreak" their devices; we figured that potential participants would be unwilling to do so.

[4] The laptop used for benchmarking was a Dell XPS 9370 laptop with an Intel Core i7-8550U CPU and 16.0 GB of RAM while the desktop device was a custom build PC with an Intel Core i7-8700 CPU with 16.0 GB of RAM and liquid cooling.

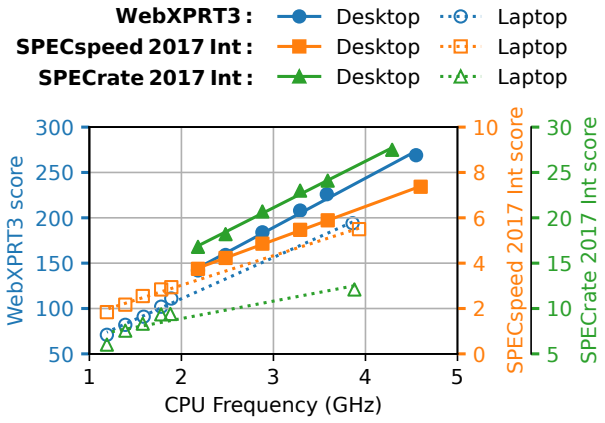


Figure 1: Using two devices and three benchmarks, we find that CPU frequency strongly correlates with performance benchmark scores, making it a reasonable proxy for device performance and an appropriate method for throttling performance in our experiments.

given the need to reliably throttle participants' devices remotely, we choose to use CPU frequency throttling in our experiments.

3 METHODS

We now describe the methodologies used in our two experiments. In all experiments, participants were recruited from Amazon's Mechanical Turk platform and were required to be at least 18 years old and working in the United States. All experimental protocols were reviewed and approved by our IRB.

3.1 Experiment #1: Device Lifetime WTA

Our first experiment subjects participants to temporary performance losses on their personal device and then elicits the *device lifetime WTA*, or the amount of money the participant would have to be paid to permanently accept the performance loss on their device. Device lifetime WTA is a measure of user resistance to permanent performance losses (such as performance losses caused by security patches to Spectre [14] and Meltdown [16]).

3.1.1 Experimental Protocol. Participants are provided with a program and are instructed to run it on their personal Windows 10 computing device. A pop-up window appears on the participants' screen which serves as the main interface for participation. After first gaining consent from participants, the program informs participants that it will test unspecified "features" on their devices by making some temporary system modifications, but does not specifically inform participants that their device's performance will be throttled during participation[5]. At this point, the program also tests its ability to actually throttle CPU frequency. Participants whose devices cannot be slowed down by the desired amount are removed from further participation.

The program then instructs participants to complete three sets of highly similar tasks (henceforth referred to as "Task 1", "Task 2", and "Task 3"). The three tasks consist of a series of subtasks chosen as a best-effort approximation of typical device usage for typical users and are nearly identical to one another (only the specific queries requested in subtasks 2, 3, and 4 below change between tasks). The subtasks are as follows:

(1) Open Microsoft Word and create a new document.
(2) Open a web browser and find the distance in miles between two specific cities and add this number to the Word document.
(3) Use the browser to find an image of a specific well-known landmark and add this image to the Word document.
(4) Use the browser to find a video of a specific live music performance on YouTube, and add the URL to the Word document.
(5) Export the Word document to a PDF and upload it to a webpage at a provided URL.
(6) Close Microsoft Word and close the web browser.

We chose to give participants a prescriptive list of tasks (rather than simply letting participants interact with a throttled device for a fixed timespan) to ensure that the participants would actually experience the performance slowdowns on their device, and to ensure a consistent experience between participants. Our chosen subtasks require participants to interact with their system in a variety of ways (opening multiple programs, using a web browser, typing, loading webpages, playing video content, etc.) in an effort to put pressure on different systems components that might be affected by frequency throttling.

Participants complete these three tasks back-to-back. Before each task, the program informs participants that some unspecified "features" will be applied for the duration of the task. Unbeknownst to the participant, the program silently throttles performance during either Task 2 or Task 3 (determined randomly with equal odds) by capping CPU frequency by either 10%, 20%, or 30%. Device performance is unthrottled during the other two tasks. To confirm that we achieve the desired slowdown, the program takes samples of the participant's device's CPU frequency during each of the three tasks and reports the samples to us in a log file (we analyze frequency samples offline and remove any participants whose devices are not throttled by the targeted amount). After performing all three sets of tasks, the program reveals to the participant that the unspecified "features" were actually throttled device performance. The program also reveals which task was throttled and by how much.

We now explain the reasons for designing our experimental protocol in this manner: First, we felt it to be necessary to expose participants to throttled and unthrottled performance back-to-back and under as similar of conditions as possible. This explains the high degree of similarity between tasks, since it gives participants the best chance at being able to fairly reflect on the differences between throttled and unthrottled performance during the subsequent exit survey. However, it is reasonable to assume that after completing Task 1, participants may "learn" the pattern of the tasks and complete subsequent tasks in less time; to avoid the possibility

[5]This slight deception gives us further informed consent (as much as is possible) to temporarily modify participants' devices (necessary for throttling performance) without specifically priming participants to think about or notice performance during the subsequent phases of the experiment. This was cleared with our IRB.

of this effect influencing participants' perceptions, we use Task 1—which is never throttled—as a "warm up" task designed simply to expose participants to the nature of the tasks (Task 1 also serves to warm up the participant's device itself, e.g. loading system caches). Additionally, we were unsure whether the ordering of the throttling (i.e. fast-then-slow vs. slow-then-fast) would have an effect on participant experience, hence the need to randomly throttle either Task 2 or 3. We also chose to not inform participants of the true purpose of the experiment beforehand so that they could interact with a throttled device without first priming them to form a mental bias on what the experience might be like.

After the Task 3 is completed, the program restores the participant's device to its pre-experiment state and guides the participant through an exit survey. The exit survey is a series of yes/no questions designed to elicit the lowest amount a participant would have to be paid to be willing to permanently accept the slowdown they just experienced on their personal device. We find this minimum dollar amount via a simple variation of exponential search: the program first asks participants if they would accept an offer to slow down their device by some percentage in exchange for \$0. If they accept, there is no lesser minimum and the willingness to accept (WTA) is \$0; otherwise the offer price is raised to \$1 and the question is asked again. If the participant declines again, the offer price is doubled to \$2, then to \$4, and so on for each time the participant declines the offer. If the participant declines the offer 15 times in a row (i.e. an offer of \$16,384), we cap the participant's WTA as "greater than \$16,384"[6]. Otherwise, there is an offer price p at which the participant *would* accept the money for the performance loss and a price $p/2$ at which the participant *would not* accept the money for the performance loss, and hence the participant's minimum WTA lies somewhere in between. The offer price is then lowered to $(p + p/2)/2$, or halfway between a known accepted offer price and a known rejected offer price. Standard binary search then commences, with each offer acceptance setting a new upper bound on the WTA and each offer rejection setting a new lower bound on the WTA. Binary search eventually converges on the participant's minimum WTA. Binary search stops when the difference between the upper and lower bounds converges to \$2, and the WTA is returned as $[(\text{upper bound}) - (\text{lower bound})]/2$.

After the WTA has been elicited, the program uploads the participants' responses to a server. Participants are then paid a flat rate for their participation. The entire process for participants takes about 20–30 minutes.

3.1.2 Replicating via Large-Scale Online Survey. We repeated the above survey questions in a large-scale online survey. This was motivated by a desire to 1) achieve a greater sample size, and 2) determine whether or not a survey mechanism, without hands-on experience with throttled performance, is an appropriate research methodology for this line of research work. Participants in the survey did not download any program and did not complete any tasks under throttled device performance, and were only asked to answer the series of yes/no questions designed to elicit WTA. Unlike the hands-on study participants (from which we elicit their WTA for only a single slowdown percentage), we elicit survey participants' WTA for slowdowns of 10%, 20%, *and* 30% in three separate series of yes/no questions. Finally, to improve data quality from survey responses, we add an "attention check" survey question[7] and remove any participants who fail the attention check. To remove additional low-quality responses, we also remove any responses where WTAs for 10%, 20%, and 30% are non-monotonic[8].

3.2 Experiment #2: Per Day WTA

A shortcoming of Experiment #1—and most surveys, for that matter—is that respondents face no consequences if they give low-quality or thoughtless responses to questions. Although we try to minimize this risk via so-called "attention check" questions, there is still little incentive for participants to deeply reflect on how much performance is worth to them. That is, both the hands-on study and the online survey lack *incentive compatibility*, meaning that participants are not incentivized to respond according to their true preferences. Thus for our second experiment, we designed and implemented an incentive compatible methodology where participants are offered the choice to throttle their personal device's performance for several days at a time in exchange for money.

Unlike Experiment #1, which elicits device lifetime WTA, our second experiment elicits the *per day WTA*, or the amount of money a participant would have to be paid to accept the experienced performance loss on a day-by-day basis. This measurement is more appropriate when determining the cost of performance losses that are temporary or reversible (e.g. optional security features like disabling hyperthreading to prevent speculation attacks). Finding the per day WTA (as opposed to device lifetime WTA) was also a necessary consequence of using an incentive compatible methodology[9].

3.2.1 Experimental Protocol. Participants are provided with a program and are instructed to run it on their personal Windows 10 computing device. A pop-up window appears on the participants' screen which serves as the main interface for participation. The program gains consent from the participant to participate and asks 1) if the program they are currently using is their primary computing device, and 2) the average number of hours each week that they use the current device. Participants who either are not using their primary device or who do not use their device for at least 10 hours a week are removed the study; we do this to ensure that potential participants would face actual consequences if their device's performance were to be throttled. Next, participants are asked to consent to having their device's clock speed monitored throughout participation and are asked to consent to using the device at least three-quarters of their average weekly hours as previously reported (also to ensure that participants face actual consequences

[6] We choose the cutoff point to be \$16,384 since it is several times higher than a brand new, high-end personal device. In addition, we add that due to our use of exponential search, we must necessarily set *some* upper limit, or else the algorithm may never terminate.

[7] Essentially a question that asks "If you are paying attention, please select the third response below"

[8] For example, suppose a participant states they would accept a 10% slowdown \$100 but also a 20% slowdown for \$10. This is illogical and demonstrates a carelessness or thoughtlessness that warrants removal. To prevent participants from accidentally giving such low-quality responses, we give survey participants the opportunity to re-do the device lifetime WTA questions if they so choose.

[9] Finding the per day WTA in an incentive compatible manner allows the experiment to conclude within a fixed timespan; finding device lifetime WTA in an incentive compatible manner would entail that the experiment last for the entirety of the participant's device's lifetime!

if the device is slowed down). At this point, the program collects the participants' device's current and maximum CPU frequency so that we can ensure that the subsequent performance throttling is successfully achieved.

Next, participants are given the opportunity to accept some dollar amount $X \in [\$0..\$10]$ in exchange for a performance loss of $N \in \{10\%, 20\%, 30\%\}$ for 24 hours. If the participant declines the offer, they are paid a small baseline compensation amount and are removed from further participation.

However, if the participant accepts the offer, the program throttles their device's max frequency by the agreed upon amount for 24 hours. The program then goes to sleep. After 24 hours have elapsed, the program wakes up and again offers participants the chance to either restore full performance or accept another $\$X$ in exchange for yet another 24 hours of throttled performance. This process continues until either the participant eventually declines to accept the money, or until the experiment times out[10]. At this point, performance is fully restored and participants are compensated their accrued earnings plus the baseline participation amount. This concludes the experimental protocol.

3.2.2 Filtering Out Invalid Results. We take several measures to ensure the validity of the data we collect in the above protocol. First, we must remove participants who decline the offer for any reason that is not strictly related to the offer price. For example, a participant may decline the offer because they do not trust us to safely throttle performance, or perhaps they share their device with others users who may not want to participate in a longitudinal study. To do this, we ask participants in an exit survey why they declined their offer and filter out all participants who report declining for any reason other than the offer amount not being high enough. This ensures that the remaining participants make decisions purely based on the monetary value of the offer itself and not for any extraneous reasons.

We also take special care to remove from consideration any participants who may try to "cheat" the experiment. For example, a cheating participant may accept the slowdown for the money but then later attempt to restore their device's performance manually. To detect any such cheaters, our program periodically audits device CPU frequency to ensure that the throttled CPU frequency has not been tampered with. Specifically, our program measures device CPU frequency the first time it is run as well as each additional day (immediately before prompting the participant to answer the WTA question). Even if such cheating goes undetected, we also ask participants if they cheated during the exit survey and offer amnesty (meaning we promise to issue payments even if they did)[11].

Participants might also try to cheat by installing the program on a spare device in an attempt to avoid enduring throttled performance. To mitigate this, we compare the timestamps from the above audit logs to participants' self-reported typical usage (in terms of hours per week, as collected during the enrollment period) to detect cheating[12]. Participants with anomalous timestamps that do not match expected usages are removed from consideration.

After these various measures have been employed, we reasonably conclude that the responses from remaining participants represent an accurate sample of participants' value of performance.

3.2.3 Replicating via Large-Scale Online Survey. As with the previous experiment, we replicate above experiment in a large-scale online survey. The motivation is again to achieve a greater sample size and determine whether or not hands-on experience is necessary for this line of research. Survey participants do not download any program and do not make any choices with real-world impact (i.e. there is no incentive compatibility). Instead, survey participants are merely asked if they would accept a dollar amount $X \in [\$0..\$20]$ in exchange for a performance loss of $N \in \{10\%, 20\%, 30\%\}$ for 24 hours[13]. To improve results, we also use an "attention check" question to filter out participants who give thoughtless responses to survey questions.

4 RESULTS

We now present the results from both sets of experiments. We reserve extended discussion and commentary for Section 5.

4.1 Experiment #1 Results

Experiment #2 was conducted for slowdowns of 10%, 20%, and 30%, for both the incentive compatible study (N=21, N=24, and N=22, respectively) and the online survey (N=306). Results from both experiments are plotted as histograms in Figure 2.

In all cases, we find that WTA approximates a power law distribution. That is, the majority of responses are grouped towards the lefthand side of the histogram (indicating a WTA of hundreds of dollars or less) while remaining responses follow a long-tailed decaying distribution. At the very end of the tail is a "bump" at the $16,384 mark, which was the maximum response possible (for reasons discussed in Section 3.1). To prevent this bump (caused by our own imposed limit) from influencing summary statistics, we choose to characterize the distributions by their median value.

From our simulation study, we find that median WTA for a performance loss of 10%, 20%, or 30% is $381, $457, and $853, respectively. From the online survey, we find that median WTA for a performance loss of 10%, 20%, and 30% is $499, $1,214, and $3,723, respectively.

We observe two trends in the data: First, we observe that, in both the study and the survey, the median WTA increases with the degree of slowdown. That is, participants would require to be paid more money in order to accept increasingly larger losses to device performance. This is an intuitive and expected result.

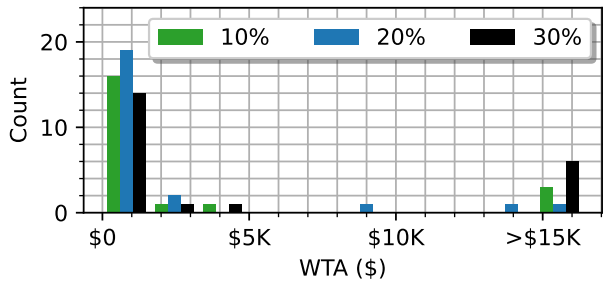
Second, we observe that the median WTAs collected from the study were across the board lower than the median WTAs collected from the online-only survey. Although the results are close for

[10] We used a cutoff of 7 days for the 10% and 20% slowdowns and 14 days for the 30% slowdown.

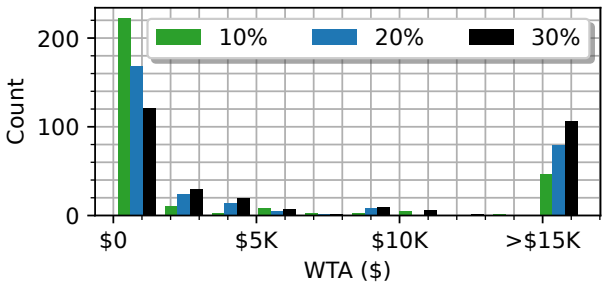
[11] No cheating was self-reported, but we did detect a few cases of cheating by looking at the audits, and removed these participants' data from consideration.

[12] For example, consider a participant who reports that they use their device for 60 hours per week, but audit timestamps are collected only once every few days. Such a participant is clearly not using their device for 60 hours a week, raising suspicions that the program may have been installed on a spare device.

[13] We chose to use a larger range in the large-scale survey ($0 to $20) than the incentive compatible study ($0 to $10). We did this to capture a wider sample range and mitigate the potentially noisier data caused by the lack of incentive compatibility or the lack of hands-on experience.



(a) Simulation study results. The median WTA for a permanent device performance loss of 10%, 20%, and 30% is $381, $457, and $853, respectively.



(b) Online survey results. The median WTA for a permanent device performance loss of 10%, 20%, and 30% is $499, $1,214, and $3,723, respectively.

Figure 2: Results from Experiment #1. We plot histograms of the data collected using the exponential yes/no elicitation mechanism (described in Section 3.1) for both the hands-on simulation study and the online survey. All distributions exhibit a sharp exponential decay followed by a bump at the >$15K mark. Although this may give the data the appearance of a "bathtub" curve, this is simply an artifact of limiting responses to be no more that $16,384 (for reasons discussed in Section 3.1).

the 10% slowdown ($381 versus $499), the results become greatly disparate by the 30% slowdown ($853 versus $3,723). Why might this be? Our hypothesis is that in the hands-on study, where participants actually experienced throttled performance on their personal device, participants were better calibrated to put a price on how much the loss in performance is actually worth. Indeed, the very reason we chose to conduct the study in the first place was because we doubted that online survey participants would be able to accurately put a price to the cost of performance losses without experiencing performance losses firsthand. For example, is a 30% loss to performance a mild annoyance, or a device death sentence? Only the study participants would be able to reasonably answer such questions. Thus a plausible explanation for the gap in the study and survey WTAs is that, after experiencing throttling firsthand, the study participants realized that the performance throttling was not as bad as they may have expected it to be.

4.2 Experiment #2 Results

Experiment #2 was conducted for slowdowns of 10%, 20%, and 30%, for both the incentive compatible study (N=26, N=29, and N=30, respectively) and the online survey (N=306). In both cases, we split participants into two categories: the "accept" group, who accept the throttled performance for the duration of their participation, and the "decline " group, who do not. The "decline" group includes participants who immediately reject the offer, as well and participants who may initially accept their offer only to later reject it before the participation period ends. The "accept" group values the amount performance lost *less* than the amount offered to them (since they accepted the offer), while the "decline" group values the performance loss *more* than the amount offered to them. Results are plotted in Figure 3.

To summarize the data and find the per day WTA, we use logistic regression to find the dollar amount at which it becomes more likely than not that a participant will accept their given offer. To do this, we first model the "decline" group as a 0 and the "accept" group as a 1. We then use the STAN modeling framework to fit a logistic curve to the data which models the probability of a participant accepting an offer given the offer price $\$x$. To summarize the data, we find the offer price x at which

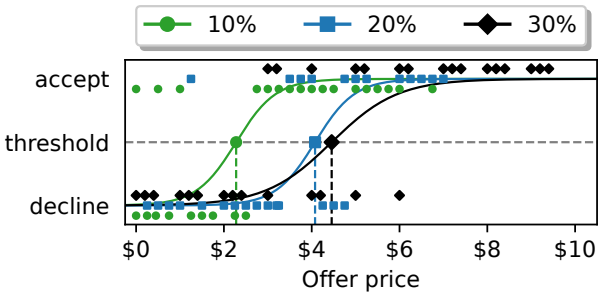
$$p(\text{outcome} = \text{“accept”} | \text{offer} = \$x) \geq 0.5$$

i.e. the point at which it is more likely than not that a participant will accept a given offer price, which we call the threshold value. From the incentive compatible study, we find the threshold values to be $2.27 per day, $4.07 per day, and $4.44 per day for slowdowns of 10%, 20%, and 30%, respectively. At the 95% confidence level, the threshold values are between $1.54 and $3.02 for a 10% slowdown, between $3.39 and $4.74 for a 20% slowdown, and between $3.40 and $5.50 for a 30% slowdown. From the online survey, we find the threshold values to be $11.15 per day, $22.85 per day, and $24.92 per day, respectively. At the 95% confidence level, the threshold values are between $7.66 and $15.38 for a 10% slowdown, between $14.69 and $41.40 for a 20% slowdown, and between $17.31 and $39.70 for a 30% slowdown. These curves are plotted in Figure 3.

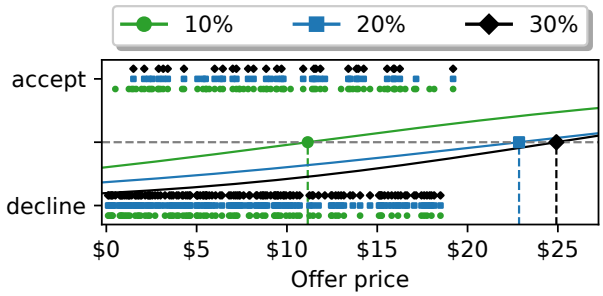
From the data, several trends emerge. First, we find that as the offer price increases, so does the likelihood of a participant accepting their offer. This is an intuitive and expected result.

Second, in the experience-based study—where participants interact with throttled device performance for several days at a time—we find that collected WTAs are once again lower than the results obtained via the online survey. Like with Experiment #1, a likely explanation is that the hands-on experience with a throttled device calibrates participants to accurately price the cost of performance losses, whereas survey participants are given no such calibration.

Third, we observe that the logistic curves are not as evenly spaced as perhaps expected: In both the study and survey data, the threshold values for the 20% performance loss are much closer to the threshold values for 30% performance losses than for 10% performance losses. While perhaps due to sampling issues, the explanation may also be that users view the harms of performance losses non-linearly. However, we also point out that this trend was not observed in Experiment #1. More experimentation is needed to definitively explain this observation.



(a) Incentive compatible study results. At offer prices of $2.27 per day, $4.07 per day, and $4.44 per day, it becomes more likely than not that a participant will accept a 10%, 20%, and 30% performance loss, respectively.



(b) Non-incentive compatible online survey results. At offer prices of $11.15 per day, $22.85 per day, and $24.92 per day, it becomes more likely than not that a participant will accept a 10%, 20%, and 30% performance loss, respectively.

Figure 3: Results from Experiment #2, which finds participants' willingness to accept per day performance losses in an incentive compatible manner. Each marker (i.e. green circle, blue square, and black diamond) represents a unique response to our offer to throttle device performance by $N \in \{10\%, 20\%, 30\%\}$ in exchange for $X per day.

Finally, we observe that the survey data is far noisier than the study data: In Figure 3a, the fitted logistic curves exhibit a fairly sharp rise from "decline" to "accept" and fairly tight confidence bounds (a couple dollars or less), whereas the survey data in Figure 3b exhibits a very slow rise and confidences bounds of tens of dollars or more. This is in spite of the survey having roughly ten times the amount of data! Once again, the likely explanation is that the lack of hands-on experience and incentive compatibility comes at a cost to precision in addition to accuracy.

5 DISCUSSION AND APPLICATIONS

We now discuss some of the implications and applications of this work.

Pricing Patches and Rebates: In the many industries (such as automotive and food, among others) there are well-established precedents, norms, and regulations for issuing product recalls and rebates when products are found to be defective or unsafe. At present, very few such precedents, norms, or regulations exist in the domain of computer systems. One recent exception was the so-called "Batterygate" affair, where Apple was found to have throttled older iPhones (allegedly to incite users into buying new phones) [21] and, after a class action lawsuit, agreed to pay consumers roughly $25 per affected device [1]. Court documents show that this settlement was reached by supposedly analyzing the resale market; however, to our knowledge, the settlement did not consider or study the impact of performance losses on actual users. In fact, our results suggest that $25 worth far less than even minor performance losses[14]. By experimentally measuring user's value of performance, we open the door to more equitably determining such rebates.

Quantifying the cost of patches becomes especially important given the rise of hardware vulnerabilities like Spectre [14] and Meltdown [16], which when patched cause significant losses to performance. It is not unreasonable to consider such critical security flaws (especially Meltdown) to be product defects, and if the trends continue, consumers may demand to have performance losses recouped in the form of rebates. Our work provides a quantitative basis for determining the harm done to users when their devices lose performance.

Implications for Security: One of the reasons why longstanding yet fixable security issues like memory safety, Spectre v-1, and Rowhammer [13] persist is at least partially because security—which users cannot quantify or evaluate—comes at the expense of performance—which users *can* quantify and and evaluate—and product vendors are reluctant to trade performance for security if it makes their products appear less valuable in the marketplace [10]. In terms borrowed from economics, this is a marketplace failure due to information asymmetry—users do not fully understand the products they purchase and product vendors hence sell to consumers' perceived needs rather than their actual needs [3], preventing the adoption of security features unless the overheads are razor-thin.

Our results suggest that perhaps there is more overhead for security than previously thought. We point to two observations: First, during the incentive compatible study, all but one of the participants who accepted their offer participated for the full duration of the experiment. This indicates that, after accepting the first offer, participants' experience with throttled performance was, in general, not worse than expected. Second, the WTAs as found in the hands-on simulation study and incentive compatible study were both lower than the counterpart WTAs as found by the online surveys. Combined, this suggests that participants' high resistance to performance losses is more psychological than based on actual needs. While our results are for end users only (and not for server-class devices where perhaps customer information asymmetry is lower and the need for performance greater), the takeaway is that user resistance to performance losses due to patches and security updates may be artificially holding back the deployment of security.

Balancing Security Tradeoffs: As previously mentioned, our experiments provide the first known effort to find the "exchange rate" between performance and user satisfaction. We now demonstrate how this exchange rate can help systems designers and architects quantitatively balance competing demands for security and

[14]We point out that our experiments were conducted on desktop and laptop users rather than phone users, but we believe the argument here still stands.

performance: Suppose that a product contains a security vulnerability that, due to ransomware, costs users an average of $1000 per year, and that the average device lifespan is two years. Now suppose that architects develop a patch for the vulnerability, but that the patch incurs a 30% performance overhead. Is this a worthwhile tradeoff? Our results suggest no: Users require at least $4.43 per day to accept a 30% performance loss, or $3241.20 across a two-year span, which is a higher than the expected $2000 losses due to ransomware over the same two-year span. Now, suppose that architects improve the patch to incur only a 10% performance overhead. According to our results, such a defense "costs" users an average of $2.27 per day, or $1657.10 over a two year span, which is lower than the expected loss to ransomware; therefore, this new defense provides more protection than it costs, in terms of user value. Thus by finding the exchange rate between performance and user satisfaction, we provide architects with a novel user-centered metric that goes beyond traditional metrics like power, performance, area, and reliability.

Are Surveys a Useful Method for Architecture Research? A secondary research goal of this work was to answer, *are surveys a worthwhile method for user-centered architecture research?* Our findings suggest not. In both experiments, the hands-on studies yielded much lower WTAs than the counterpart online surveys. This supports our belief that users are not sufficiently knowledgeable or experienced with device performance to be able to make decisions that accurately reflect their true preferences. Unfortunately, this means that user-focused architecture research requires experimentation rather than relying on simple and easy-to-deploy surveys. Our work provides a template for future researchers on how to design and build such experiments.

Cross-Validating the Experiments: Another observation we make is that the results of the two sets of experiments may appear to be somewhat incongruous: Users would accept at 10% performance loss for $381 but also $2.27 per day, putting the "break even" point between the two results at roughly half a year[15]. If participants responded rationally in both cases, we might expect this "break even" point to be closer to device lifetime, which is almost certainly longer than half a year. What might this be? We offer two possible explanations: First, the simulation study is not incentive compatible, allowing for the possibility that the study participants in Experiment #1 did not answer the WTA question with as much thought and attention as it may have deserved. Another likely explanation is simply that humans are not perfectly rational when reasoning about small amounts (i.e. per day WTAs) vs. long-term events (i.e. device lifetime performance losses).

6 RELATED WORK

User-Centered Design: In the architecture and systems communities, the end user is often seen as being many layers of abstraction removed from the hardware, and thus the end user oftentimes is not considered during the hardware and systems design process. Our work and other similar lines of research attempt to cut through these layers of abstraction by providing user-centered metrics to better aid user-focused system design. Several user studies have leveraged dynamic voltage and frequency scaling (DVFS) to help balance the competing demands of energy efficiency and user satisfaction [18, 24–26]. Individualized quality of service (QoS) metrics have also been proposed as a means towards achieving this balance [29–31]. Other work identifies the components and design configurations that yield higher user satisfaction [9, 28].

User-centered metrics for improving user satisfaction have not been confined to academia: Intel's Project Athena has introduced metrics for its EVO line of laptops called Key Experience Indicators (KEIs) that quantify elements of the user experience, such as wake time, responsiveness, and charging times [27].

Incentive Compatible Mechanisms: While the above work on user-centered design is thematically similar to our own, the methodologies do not employ incentive compatible study designs. Our work aims to raise the bar for user studies for hardware design by introducing incentive compatible methodologies and mechanisms. We found inspiration from prior incentive compatible studies [5, 6] which use incentive compatible mechanisms like BDM lotteries [4], best-worst scaling (BWS) [17], and the mechanism used in our own experiments, the single discrete binary choice mechanism [7].

Pricing Performance: Measuring the monetary value of performance has been attempted before but from the perspective of businesses rather than end users. Prior studies have found that increases to latency hurt e-commerce revenue [15, 23], largely because keeping users engaged with services requires low latency [20]. The value of latency has also been studied in financial markets and high-frequency trading [19, 22]. To our knowledge, our work is the first of its kind to put a price to end user's value of performance.

7 CONCLUSION

We are in the midst of a revolution in computer architecture and computer hardware design [11]. The huge demand for vertically integrated products along with the rise of open source hardware has pushed hardware companies to rapidly innovate to create products that must meet high demands of integration, performance, energy efficiency, and cost. In the push to meet the changing design requirements, we introduce a new metric—willingness to accept performance losses—that defines the "exchange rate" between user satisfaction and system performance. This new metric lets systems designers and architects to quantitatively consider the dollar cost, from the users' perspective, of performance-costing features when designing future systems.

To aid in this new paradigm, we present the first work on hardware behavioral economics. Two methodologies are used to elicit users' willingness to accept performance losses. Our first experiment finds that users would accept a permanent performance loss of 10%, 20% and 30% on their personal device in exchange for $381, $457, and $823, respectively, while a larger-scale online-only survey finds the same results to be $499, $1214, and $3723, respectively. Our second experiment finds that users would accept, per day, a performance loss of 10%, 20% and 30% on their personal device in exchange for $2.27, $4.07, and $4.43 per day, respectively, while a larger-scale survey finds the same results to be $11.15, $22.85, and $24.92 per day, respectively.

[15] I.e. after half a year, the value of receiving a daily $2.27 payment exceeds a flat rate payment of $381.

REFERENCES

[1] 2020. IN RE: APPLE INC. DEVICE PERFORMANCE LITIGATION. https://www.smartphoneperformancesettlement.com/docs/Supplemental_Joint_Declaration_of_Joseph_W_Cotchett_and_Laurence_D_King_in_Support_of_Motions.pdf

[2] Akamai. 2017. The State of Online Retail Performance.

[3] Ross Anderson. 2001. Why information security is hard-an economic perspective. In *Seventeenth Annual Computer Security Applications Conference*. IEEE, 358–365.

[4] Gordon M Becker, Morris H DeGroot, and Jacob Marschak. 1964. Measuring utility by a single-response sequential method. *Behavioral science* 9, 3 (1964), 226–232.

[5] Erik Brynjolfsson, Avinash Collis, and Felix Eggers. 2019. Using massive online choice experiments to measure changes in well-being. *Proceedings of the National Academy of Sciences* 116, 15 (2019), 7250–7255.

[6] Erik Brynjolfsson, Felix Eggers, and Avinash Gannamaneni. 2018. Measuring welfare with massive online choice experiments: A brief introduction. In *AEA Papers and Proceedings*, Vol. 108. 473–76.

[7] Richard T Carson, Theodore Groves, and John A List. 2014. Consequentiality: A theoretical and experimental exploration of a single binary choice. *Journal of the Association of Environmental and Resource Economists* 1, 1/2 (2014), 171–207.

[8] Dennis F Galletta, Raymond Henry, Scott McCoy, and Peter Polak. 2004. Web site delays: How tolerant are users? *Journal of the Association for Information Systems* 5, 1 (2004), 1.

[9] Matthew Halpern, Yuhao Zhu, and Vijay Janapa Reddi. 2016. Mobile CPU's rise to power: Quantifying the impact of generational mobile CPU design trends on performance, energy, and user satisfaction. In *2016 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. 64–76. https://doi.org/10.1109/HPCA.2016.7446054

[10] Adam Hastings and Simha Sethumadhavan. 2020. WaC: A New Doctrine for Hardware Security. In *Proceedings of the 4th ACM Workshop on Attacks and Solutions in Hardware Security*. 127–136.

[11] John Hennessy and David Patterson. 2018. A New Golden Age for Computer Architecture: Domain-Specific Hardware/Software Co-Design, Enhanced Security, Open Instruction Sets, and Agile Chip Development. Turing Award Lecture.

[12] John A Hoxmeier and Chris DiCesare. 2000. System response time and user satisfaction: An experimental study of browser-based applications. (2000).

[13] Yoongu Kim, Ross Daly, Jeremie Kim, Chris Fallin, Ji Hye Lee, Donghyuk Lee, Chris Wilkerson, Konrad Lai, and Onur Mutlu. 2014. Flipping bits in memory without accessing them: An experimental study of DRAM disturbance errors. *ACM SIGARCH Computer Architecture News* 42, 3 (2014), 361–372.

[14] Paul Kocher, Jann Horn, Anders Fogh, Daniel Genkin, Daniel Gruss, Werner Haas, Mike Hamburg, Moritz Lipp, Stefan Mangard, Thomas Prescher, et al. 2019. Spectre attacks: Exploiting speculative execution. In *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 1–19.

[15] Grey Linden. 2006. Guest class lecture, CS345 (Data Mining). http://sites.google.com/site/glinden/Home/StanfordDataMining.2006-11-28.ppt Stanford University.

[16] Moritz Lipp, Michael Schwarz, Daniel Gruss, Thomas Prescher, Werner Haas, Stefan Mangard, Paul Kocher, Daniel Genkin, Yuval Yarom, and Mike Hamburg. 2018. Meltdown. *arXiv preprint arXiv:1801.01207* (2018).

[17] Jordan J Louviere, Terry N Flynn, and Anthony Alfred John Marley. 2015. *Best-worst scaling: Theory, methods and applications*. Cambridge University Press.

[18] Arindam Mallik, Jack Cosgrove, Robert P. Dick, Gokhan Memik, and Peter Dinda. 2008. PICSEL: Measuring User-Perceived Performance to Control Dynamic Frequency Scaling. In *Proceedings of the 13th International Conference on Architectural Support for Programming Languages and Operating Systems* (Seattle, WA, USA) *(ASPLOS XIII)*. Association for Computing Machinery, New York, NY, USA, 70–79. https://doi.org/10.1145/1346281.1346291

[19] Ciamac C Moallemi and Mehmet Saglam. 2010. The cost of latency. *SSRN eLibrary* (2010).

[20] Jakob Nielsen. 1994. *Usability engineering*. Morgan Kaufmann.

[21] John Poole. 2017. *iPhone Performance and Battery Age*. https://www.geekbench.com/blog/2017/12/iphone-performance-and-battery-age/

[22] Ryan Riordan and Andreas Storkenmaier. 2012. Latency, liquidity and price discovery. *Journal of Financial Markets* 15, 4 (2012), 416–437.

[23] Eric Schurman and Jake Brutlag. 2009. Performance Related Changes and their User Impact. https://www.youtube.com/watch?v=bQSE51-gr2s O'Reilly Velocity 2009 Conference.

[24] Alex Shye, Berkin Ozisikyilmaz, Arindam Mallik, Gokhan Memik, Peter A. Dinda, Robert P. Dick, and Alok N. Choudhary. 2008. Learning and Leveraging the Relationship between Architecture-Level Measurements and Individual User Satisfaction. In *2008 International Symposium on Computer Architecture*. 427–438. https://doi.org/10.1109/ISCA.2008.29

[25] Alex Shye, Yan Pan, Ben Scholbrock, J. Scott Miller, Gokhan Memik, Peter A. Dinda, and Robert P. Dick. 2008. Power to the people: Leveraging human physiological traits to control microprocessor frequency. In *2008 41st IEEE/ACM International Symposium on Microarchitecture*. 188–199. https://doi.org/10.1109/MICRO.2008.4771790

[26] Alex Shye, Benjamin Scholbrock, and Gokhan Memik. 2009. Into the wild: Studying real user activity patterns to guide power optimizations for mobile architectures. In *2009 42nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. 168–178. https://doi.org/10.1145/1669112.1669135

[27] Jim Scovell Sudha Ganesh and Clem Wong. 2020. Measuring What Matters: Project Athena Innovation Program's Real-World Testing. https://www.intel.com/content/www/us/en/products/docs/devices-systems/laptops/laptop-innovation-program/real-world-testing.html

[28] Kaige Yan, Jingweijia Tan, and Xin Fu. 2019. Bridging mobile device configuration to the user experience under budget constraint. *Pervasive and Mobile Computing* 58 (2019), 101023. https://doi.org/10.1016/j.pmcj.2019.05.004

[29] Kaige Yan, Jingweijia Tan, and Xin Fu. 2019. Improving energy efficiency of mobile devices by characterizing and exploring user behaviors. *Journal of Systems Architecture* 98 (2019), 126–134. https://doi.org/10.1016/j.sysarc.2019.07.004

[30] Kaige Yan, Xingyao Zhang, and Xin Fu. 2015. Characterizing, modeling, and improving the QoE of mobile devices with low battery level. In *2015 48th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. 713–724. https://doi.org/10.1145/2830772.2830786

[31] Kaige Yan, Xingyao Zhang, Jingweijia Tan, and Xin Fu. 2016. Redefining QoS and customizing the power management policy to satisfy individual mobile users. In *2016 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. 1–12. https://doi.org/10.1109/MICRO.2016.7783756

A APPENDIX A: DEVICE LIFETIME WTA USER STUDY INSTRUMENT

A.1 Abstract

This artifact is the program used for Experiment #1 in the paper. As described in Section 3.1, it guides the user through a series of tasks and throttles performance during either Task 2 or Task 3

A.2 Artifact check-list (meta-information)

- **Run-time environment:** Program must be run on Windows 10 devices.
- **Hardware:** In order to view frequency throttling, the test device needs to have DVFS (dynamic voltage and frequency scaling) enabled.
- **Run-time state:** Device's frequency may be throttled during program execution.
- **Publicly available?:** Yes
- **Code licenses (if publicly available)?:** Creative Commons Attribution 4.0 International
- **Archived (provide DOI)?:** 10.5281/zenodo.7808928

A.2.1 How to access. Accessible at https://zenodo.org/record/7808928.

A.3 Installation

Extract `Experiment.zip` and run the program `run.exe` to run. A window will appear and guide the user through a series of questions and tasks. All changes made to the test device are reversed after completing participation.

A.4 Evaluation and expected results

During either Task 2 or Task 3, device CPU frequency will be capped by the percentage specified in `Files/cfg.txt`. This is not guaranteed to happen on all devices.

A.5 Methodology

Submission, reviewing and badging methodology:

- https://www.acm.org/publications/policies/artifact-review-badging
- http://cTuning.org/ae/submission-20201122.html
- http://cTuning.org/ae/reviewing-20201122.html

B APPENDIX B: PER DAY WTA USER STUDY INSTRUMENT

B.1 Abstract

This artifact is the program used for Experiment #2 in the paper. As described in Section 3.2, it gives users an offer between performance and money.

B.2 Artifact check-list (meta-information)

- **Run-time environment:** Program must be run on Windows 10 devices.
- **Hardware:** In order to view frequency throttling, the test device needs to have DVFS (dynamic voltage and frequency scaling) enabled.
- **Run-time state:** Device's frequency may be throttled during program execution.
- **Publicly available?:** Yes
- **Code licenses (if publicly available)?:** Creative Commons Attribution 4.0 International
- **Archived (provide DOI)?:** 10.5281/zenodo.7808924

B.2.1 How to access. Accessible at https://zenodo.org/record/7808924.

B.3 Installation

Extract `Experiment.zip` and run the program `run.exe` to run. A window will appear and guide the user through a series of questions and then give the offer between performance and money (the actual offer was only valid during the study period, which has passed). All changes made to the test device are reversed after completing participation.

B.4 Evaluation and expected results

If the offer is accepted, the test device will may have its CPU frequency decreased by the percentage specified in `Files/scfg.txt`. This is not guaranteed to happen on all devices. The specific offer price given is specified in `Files/ocfg.txt`.

B.5 Methodology

Submission, reviewing and badging methodology:

- https://www.acm.org/publications/policies/artifact-review-badging
- http://cTuning.org/ae/submission-20201122.html
- http://cTuning.org/ae/reviewing-20201122.html

C APPENDIX C: SURVEY INSTRUMENT

C.1 Abstract

This artifact is the web-based survey used in this work. It is a single html web form.

C.2 Artifact check-list (meta-information)

- **Publicly available?:** Yes
- **Code licenses (if publicly available)?:** Creative Commons Attribution 4.0 International
- **Archived (provide DOI)?:** 10.5281/zenodo.7809024

C.2.1 How to access. Accessible at https://zenodo.org/record/7809024.

C.3 Installation

Open in any web browser to view.

C.4 Evaluation and expected results

No evlaution needed.

C.5 Methodology

Submission, reviewing and badging methodology:

- https://www.acm.org/publications/policies/artifact-review-badging
- http://cTuning.org/ae/submission-20201122.html
- http://cTuning.org/ae/reviewing-20201122.html