

Modeling User-Centered Page Load Time for Smartphones

Conor Kelton
ckelton@cs.stonybrook.edu
Stony Brook University

Jihoon Ryoo
jiryoo@cs.stonybrook.edu
SUNY Korea

Aruna Balasubramanian
arunab@cs.stonybrook.edu
Stony Brook University

Xiaojun Bi
xiaojun@cs.stonybrook.edu
Stony Brook University

Samir R. Das
samir@cs.stonybrook.edu
Stony Brook University

ABSTRACT

Page Load Time (PLT) is critical in measuring web page load performance. However, the existing PLT metrics are designed to measure the Web page load performance on desktops/laptops and do not consider user interactions on mobile browsers. As a result, they are ill-suited to measure mobile page load performance from the perspective of the user. In this work, we present the Mobile User-Centered Page Load Time Estimator (muPLT_{est}), a model that estimates the PLT of users on Web pages for mobile browsers. We show that traditional methods to measure user PLT for desktops are unsuited to mobiles because they only consider the initial viewport, which is the part of the screen that is in the user's view when they first begin to load the page. However, mobile users view multiple viewports during the page load process since they start to scroll even before the page is loaded. We thus construct the muPLT_{est} to account for page load activities across viewports. We train our model with crowdsourced scrolling behavior from live users. We show that muPLT_{est} predicts ground truth user-centered PLT, or the muPLT , obtained from live users with an error of 10-15% across 50 Web pages. Comparatively, traditional PLT metrics perform within 44-90% of the muPLT . Finally, we show how developers can use the muPLT_{est} to scalably estimate changes in user experience when applying different Web optimizations.

ACM Reference Format:

Conor Kelton, Jihoon Ryoo, Aruna Balasubramanian, Xiaojun Bi, and Samir R. Das. 2020. Modeling User-Centered Page Load Time for Smartphones. In *22nd International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '20)*, October 5–8, 2020, Oldenburg, Germany. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3379503.3403565>

1 INTRODUCTION

It is well understood that Web page load performance, also called the Page Load Time (PLT) is critical for user engagement [11, 16]. For example, Walmart reports 1% increase in revenue for every 100ms reduction in PLT [11]. As a result, developers use several page load metrics [4, 20, 36] to measure and optimize for page performance.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

MobileHCI '20, October 5–8, 2020, Oldenburg, Germany

© 2020 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-7516-0/20/10...\$15.00
<https://doi.org/10.1145/3379503.3403565>

The critical question is: do these traditional page load metrics measure user-centered latencies? User-centered latency is the time until the user perceives that the page is loaded and they can start to engage with the page. Most existing page load metrics are designed for desktops/laptops and measure the time to load the *initial viewport*; i.e., the area of the page initially visible to the user. However, anecdotally, as less content is available in the initial viewport of smartphones, users start to *scroll* even while the content is still loading [2]. This means the page load time not only depends on the page loading on the initial viewport, but on all the viewports that a user is likely to view.

Our goal is to define, measure, and model the user-centered page load metric for mobile browsers. We call this metric **muPLT** or **mobile user-centric Page Load Time**. Figure 1 shows the need for a new metric. The figure shows the state of the page *bloomberg.com* at the PLT time as measured by the three most popular metrics—OnLoad [4], First Contentful Paint [36], SpeedIndex [20] (See Section 3 for an in-depth description of these metrics). There, these metrics are compared to muPLT that we obtain using our crowdsourcing technique.

Instead of these existing methods, we propose to design a new metric that better correlates with the user's perceived latency. We take three steps towards achieving our goal: i) to understand the user's scrolling behavior on browsers and show that mobile users view multiple viewports before perceiving a page to be loaded, ii) to collect the ground truth muPLT via extensive mobile user studies that takes into account mobile-specific interaction patterns, and iii) to develop the muPLT_{est} model that estimates the ground truth muPLT without requiring extensive user studies beyond an initial training phase.

We start by designing a new crowdsourcing platform to study mobile browser interactions. To standardize the page load process across different network and device conditions, mobile user studies usually show a video of the page load process [21, 45]. However, this means that the users cannot interact with the page in terms of scrolling¹. Instead, we create synchronized videos of multiple viewport loads and stitch them together to allow mobile users to scroll through while still keeping the page load process standardized. We show that most users do not perceive a difference between our video recording and the real page load.

We use our platform to collect the ground truth muPLT from users. We explicitly ask the users to respond when they think the page is loaded enough for them to engage with the page. However,

¹Users can also interact with page by tapping and pinching, but these interactions either cannot be responded to by the page until the page loads or cannot be responded to by mobile pages at all.

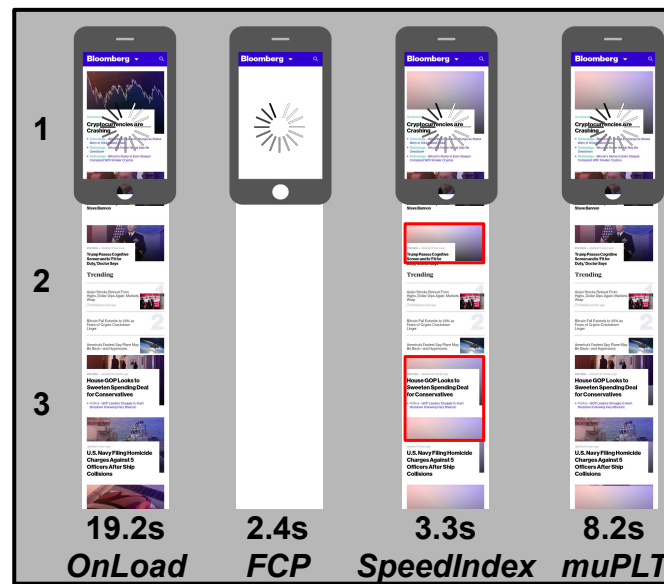


Figure 1: State of the page, *bloomberg.com* at the PLT measured by three existing page load metrics—OnLoad, SpeedIndex, and First Contentful Paint. This is compared to the user-centric metric we define, muPLT which measures the time when the user perceives the page to be load. muPLT is calculated based on user studies. The existing page load metrics differ considerably from the user-centric metric.

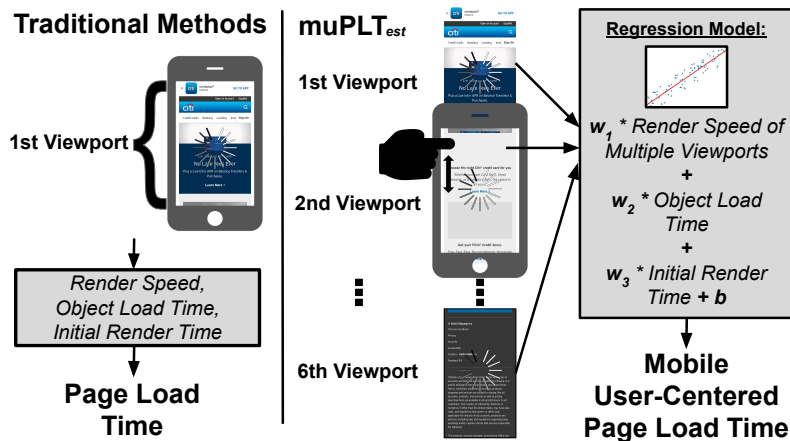


Figure 2: Most traditional Page Load Time (PLT) metrics, shown on the left, measure the page load time on the initial *viewport* of the Web page, which is the part of the Web page initially visible to users. However, the existing PLT metrics do not correlate well with the time users believe the page to be loaded, the muPLT. We present muPLT_{est}, a model that accurately measures the muPLT. muPLT_{est} takes into account mobile Web user interactions where a user scrolls through multiple viewports on the smartphone before determining if the page is loaded.

this explicit measure exhibits high variance across users, due to the *dual task paradigm* [43], where users degrade in task performance due to having to both interact with the page and provide explicit feedback. Instead, we design an implicit muPLT metric that uses fine-grained scrolling information to capture the ground truth muPLT.

Our user studies with 300 users across 50 Webpages and different network conditions show that muPLT differs by 42-70% compared to three most popular page load metrics. In other words, existing page load metrics that are traditionally used to measure mobile page performance do not accurately capture the user-centered latency in

practice. In addition, muPLT correlates well with the explicit user feedback and exhibits lower variance across users.

Finally, we design muPLT_{est} to predict muPLT . The muPLT_{est} can be estimated automatically from browser tools and accurately measures the ground truth muPLT . muPLT_{est} takes into account the page load activities on multiple viewports and exploits the observation that existing PLT metrics individually estimate different aspects of the page load process. muPLT_{est} first computes the traditional PLT metrics on multiple viewports and then combines them using a simple regression to predict muPLT . Figure 2 highlights end-to-end our approach in developing a more user-centered metric contrast to traditional approaches. muPLT_{est} is indeed able to accurately predict muPLT , with a median prediction error of 10–15% across the same 300 users and 50 pages.

The motivation for designing muPLT_{est} is to allow developers to scalably compare their page load optimizations with respect to a user-centered metric. To this end, we show how muPLT_{est} can be used by developer in practice using a case study with the Server-Push [30, 48] optimization. We show that developers can accurately evaluate different push strategies without requiring extensive user studies by using muPLT_{est} .

2 RELATED WORK

We discuss related work in the areas of mobile interactions and crowdsourced user studies.

2.1 User Interactions on Mobile Web Pages

There has been considerable work on studying how users interact with mobile devices in general, and with mobile Web pages specifically. One of the earliest works to study scrolling behavior, Shreshta et al. [39] shows that scrolling is more common than pagination for mobile Web content. Lee et al. [41] use 3D motion tracking to enable off-screen scrolling which results in higher user satisfaction when interacting with large, multi-screen applications. The authors of Spotlights [26] create a system to overlay important page content users may miss when scrolling through large mobile pages.

In addition to scrolling there has been work studying how other interactions, including tapping and pinching, can help identify problems in mobile page design [37]. However, tapping and pinching are relevant to the user experience only after the pages have fully loaded, whereas, we are interested in the time just up until the page is loaded. In fact, certain pages are known to not be even responsive to these interactions until after the page is loaded [33].

2.2 Mobile versus Non-Mobile quantification

There has also been considerable work differentiating user behavior on desktop versus mobile. Studies show that the mobile Web users perform different tasks compared to desktop Web users [14, 27]. Findlater et al. show that user interactions change between lab versus crowdsourced studies, and depend on whether the study is done on mobile or desktop devices [18]. XDBrowser [32], provides a new cross-device Web browser which can automatically translate a page design from one device type to another.

Finally, Kumar et al. develop a method to perform eye tracking studies over Web pages with multiple viewports. They track gaze as users scroll through pages and visualize the tracks on a single

stimulus of the stitched together viewports [31]. Their efforts show a shift in focus away from simply quantifying the user experience for Web pages using just the first viewport.

2.3 Crowdsourcing

Crowdsourcing has been an important tool in obtaining user data at scale [6, 23, 38]. Past crowdsourced tasks include document editing and writing [10], researching product comparisons [22], and automatic question and answer [24]. Recently, crowdsourcing has been used to better define mobile specific behaviors and interactions [8, 12, 17, 18].

A main challenge for crowdsourcing is that users operate outside of a controlled setting [18, 23]. In the context of Web page loads, this means the process can vary significantly depending on the end user's device and network condition. Related works [21, 33, 45] on non-mobile browsers have standardized the page load process by recording a video of the page load. They then let users view the videos as though they are experiencing live loads. Our work uses the same standardization technique, but also allows the user to scroll.

3 PLT METRICS AND THEIR LIMITATIONS

Page Load Time (PLT) is used to measure the time to load the contents of the page, after which the user can engage with the page. Given the importance of the PLT measure for Web development, researchers and practitioners have defined several metrics to measure PLT, as we discuss below.

3.1 Traditional PLT metrics

Resource loading based metrics The most commonly used PLT metric is *OnLoad*. *OnLoad* measures the time from when the browser receives the URL to when all Web page objects have been downloaded, parsed, and evaluated [4]. However, studies [3, 21, 33] have shown that this metric does not correlate well with the user's perceived latency because *OnLoad* given equal importance to all objects on the page, even those that are not important to the user.

Visual metrics *SpeedIndex* is a visual metric that measures the visual rendering progress of a page. However, *SpeedIndex* only considers the initial viewport, which is content that can fit in the users screen, colloquially known as above-the-fold or ATF. *SpeedIndex* is designed for desktops/laptops, where a user only takes into account the page load on the initial viewport to determine if a page is loaded; we corroborate this usage of *SpeedIndex* in our user studies.

First Contentful Paint, or *FCP* is another visual metric that measures the time for the first object to be rendered [36]. The intuition is to identify the earliest possible time at which a user can detect perceivable difference in the page load. *FCP* also only considers the initial viewport.

Time-to-interactivity (TTI) Recent studies from both academia (Vesper [33]) and industry (*Time-to-interactivity* [35]) define page load time based on when the user can interact with the content. Example interactions include clicking on a link or using the auto-complete feature of a search bar. TTI metrics also only consider the initial viewport [33]. We do not consider this metric in our study because we largely consider the type of browsing behavior that does not include such interactions with the page.

User-Centered PLT Two recent studies, WebGaze [21] and Eye-Org [45] both show that the traditional PLT metrics, notably On-Load and SpeedIndex, do not correlate with latency given by users for desktop browsers. They conduct large scale, standardized user studies to get explicit feedback on user-centered page load times. But both studies only consider the initial page viewport.

3.2 Why do we need a new metric?

In summary existing page load metrics fall into one or both cases:

- They define the page load time in terms of the initial viewport [20, 21, 33, 45] alone
- They focus on objective measures such as resource loading [4] or visual completeness [20, 36] that may differ from user perception

In this work, we define the muPLT which takes into account user interaction patterns to better capture a user-centered page load time (See Figure 1 for an explicit example). We design muPLT_{est} to accurately predict muPLT without the need for large scale user studies as is done in the current literature [21, 45, 46].

4 A NEW CROWDSOURCING PLATFORM FOR MOBILE BROWSERS

We design a crowdsourced platform to study how mobile users interact with browsers on smartphones. All studies described in this work are approved by our Institutional Review Board (IRB).

4.1 Crowdsourcing with user interaction

As is usual in crowdsourced studies for Web page loads [21, 33, 45], we standardize the Web page load process to remove variability caused by network conditions and end user devices. The common technique [21, 33, 45] for such standardization in crowdsourcing is to video record the process to show to users. We also include means to filter out responses from are idling or spamming when giving their muPLT feedback. Specifically, we reject any responses that occurred before the first, and after the last, visual changes across all viewports of the page [21]. Microworkers requires uploading a proof of completion which is given to users by our user study servers after successfully completing the study. The proof was not provided to users who had greater than 5 rejected responses. Any user who failed to upload their proof was not included in our final dataset and not compensated. These measures filtered out 18% of users.

The problem is that existing studies only video record the initial viewport, because PLT is defined only over the initial viewport. However, on small form factor devices, anecdotally, users view multiple viewports even before the page is loaded [2]. To this end, we design a study that allows the user to freely scroll through multiple viewports, while also standardizing the page load process.

Our idea is to record a video of the load process from each viewport of the page, stitch them together, and then play them back simultaneously. Figure 2 shows an example page with multiple viewports worth of content present; the viewports are numbered 1 through 6. The video recording is done until the lowest possible viewport [31]. During the study, the aspect ratio of the videos

are scaled according to the size of the participants phone. Correspondingly, all coordinates for interactions, such as scrolling, are re-scaled. We faithfully recreate Web page behavior where the page cannot be scrolled beyond a certain point until the content is fully loaded. We use HTTP Record and Replay [28, 34] to make sure the content is the same at each recorded viewport.

Recording client interaction We use client-side log analysis [19, 37] to record the user interactions. To build the logs we record the start time, end time, and distance of each scroll via JavaScript. This logging methodology is similar to prior work [37].

Observer effect A natural question is: does scrolling behavior change the way the page loads? Recall that we record videos from each viewport. If scrolling to these viewports changed how the page loads, then the user will not experience the real page load process during playback. Fortunately, we find that scrolling does not change the page load process. Our experiments (not shown here for brevity) show that OnLoad measures and the visual progression of the page load remain the same with or without scrolling.

4.2 Qualitative analysis of the crowdsourcing platform

Our crowdsourcing platform video records the full page load process and shows it to the user, as though the page is loaded live. As our studies are meant to recreate the mobile browsing experience, a key qualitative question is if the users perceive a difference between (A) viewing the live page and (B) viewing the recorded page through the crowdsourcing platform.

We conduct an A/B test across 30 users, for 25 of the pages in our experiments. Each user was shown both versions of the page, A and B, in different orders. After each pair, users were asked “Aside from the speed of the page load, did you notice anything unnatural when browsing either page?”.

On average, users found nothing unnatural about 22 of the 25 video pages, suggesting their experience with our mobile pages load videos was qualitatively similar to normal mobile browsing for the overwhelming majority of pages. The standard deviation was 3.5 pages. For the other 3 videos, on average, that were reported as unnatural browsing, 44% of comments were of the form, “I visit this site regularly, why did it look blurrier than usual?”. This is likely because the videos were taken on Nexus 5 in lower quality than the client’s native device. Additionally, 15% took the form that “The site load contained older content from a previous date”. This was because the standardized page load was recorded at an earlier date while the live site loaded the most recent version of the page.

Overall, our takeaway is that the standardized videos provide a sufficiently close mobile user experience to live Web pages during the page load.

5 CHARACTERIZING SCROLLING BEHAVIOR ON MOBILE/NON-MOBILE WEB PAGES

Using the crowdsourced set up described above, we explore how mobile users scroll through Web pages during the page load process in comparison to desktop users and under different tasks.

5.1 Study Methodology

We create Web page videos as described in the previous section under different network conditions on smartphones and desktops. The mobile network conditions emulate 3G and 4G speeds obtained from existing network emulators [7]. On the desktop, we choose a network with 50ms RTT and 15 Mb/s throughput, which corresponds to broadband ethernet speeds [33]. Videos were taken using the *screenrecord* functionality of the Android Debugging Bridge (adb) tool [40] and page loads are automated via remote controlling (debugging) Google Chrome [13].

We choose the 50 Web pages by taking a random sample stratified across the categories of the Alexa Top 500 [5]. During the Web page load, we also measure the traditional PLT metrics: OnLoad [4], SpeedIndex [20] (estimated using videos), and FCP [36].

Devices All our recordings are done on three devices: i) 1080×1920, 4.95' Google Nexus 5 phone using Google Chrome version 61.0, ii) 1440×2560, 5.5' Google Pixel XL using Google Chrome version 67.0, and iii) 1920×1080, 23' monitor on an Ubuntu 16.04 desktop using Google Chrome version 67.0.

Study platform We use the *Microworkers* platform [1] as our crowdsourcing distribution medium. The Microworkers platform allows researchers to specify the device (mobile versus non-mobile) used. The study was done across 22 countries from the regions of the Caribbean, Asia, Africa, Eastern and Western Europe and the United States.

Study Tasks We asked the users to perform two kinds of tasks: Task 1 (Free browsing): The first is a *free browsing task* similar to related work [21, 33, 45]. We ask the users: "Provide feedback using the submit button when you feel the page is loaded enough for you to engage with the page." The users can go back and re-view the page load if needed. During the experiment we track the users' scrolling behavior, but we do not tell them explicitly to scroll or mention that we are capturing this behavior. We conduct this study across 200 users and 50 Web pages in total, with 50 users for the 3G study on the Nexus 5, 50 for the 4G study on the Nexus 5, 50 users for the 4G study on the Pixel XL and 50 users on the Wifi study on Desktops. [Each user saw each page in all studies discussed, thus all analyses for each page discussed in this work are based on feedback across 50 users. These 50 users were those who passed our crowdsourcing filters.](#)

Task 2 (Targeted search): The second is a more targeted task-based study. Behavior and relative performance of users has been shown to be highly influenced by the task they are posed [15, 29]. To this end, we conduct a smaller scale experiment, 50 users across the same Web pages over a single device and network condition, where users are asked to perform the task of searching for specific objects on the loaded page. Specifically we pose the question as "Find X", where "X" is a distinguishable Web page object such as a specific image or headline on each page.

5.2 User scrolling behavior on mobile browsers

We characterize how users interact with mobile browsers based on how many viewports they view before they perceive the page to be loaded.

Number of viewports viewed Figure 3 shows a CCDF across pages showing the percentage of users who view more than the

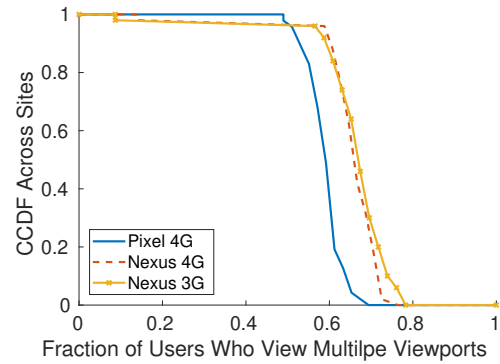


Figure 3: CCDFs of the fraction of users who view multiple viewports before they perceive the page to be loaded. Each point represents that y% of Web pages have x% of users, or greater, that scroll.

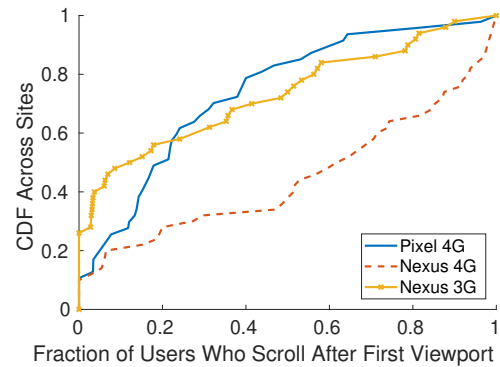


Figure 4: A CDF of fraction of users who start to scroll after the initial viewport is fully loaded.

initial viewport before the page is loaded. Over 70% of the users view multiple viewports in over 50% of the Web pages for Nexus 5, with somewhat smaller values for the Pixel. This difference can be attributed to the larger screen size of Pixel phones.

When do users start to scroll to multiple viewports? Figure 4 captures the percentage of users who start to scroll after the initial viewport is fully loaded. We use the SpeedIndex [20] to calculate the time when the initial viewport is fully loaded. For the Nexus 5, of the users who scroll on 3G Networks, only 12% of users waited for all the initial viewport to be loaded before scrolling, in the median case. Even on the faster 4G networks, 40% still scrolled before the first viewport loaded for the median page. For the Pixel phone the larger screen size means that fewer users start scrolling, however more than 80% of users on the Pixel still scrolled before the first viewport was completed for the median page.

Number of viewports viewed before page load Figure 5 shows the number of viewports on an average that a user scrolls before perceiving the page to be loaded. For the median page, the median user traverses ~ 3 viewports of content before perceiving the page

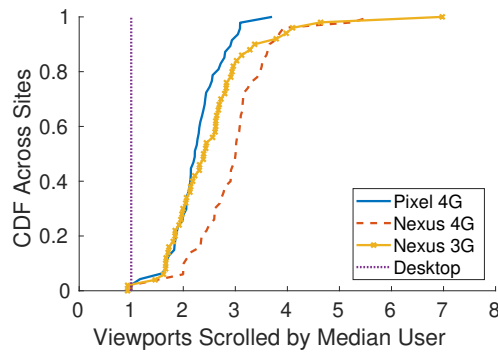


Figure 5: CDF of the number of viewports users viewed before perceiving the page to be loaded, both on desktop and mobile browsers. On the mobile browser, the majority of users look beyond the first viewport of content before determining a page to be loaded. In contrast, on desktop browsers, the median user does not view more than the initial viewport.

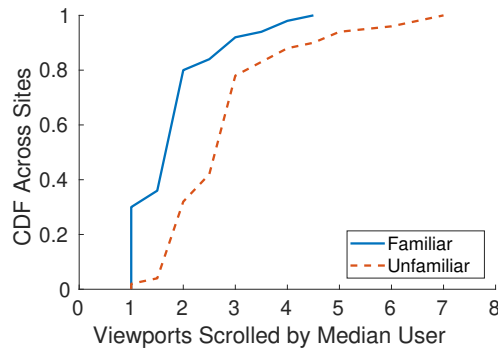


Figure 6: A CDF showing the raw number of viewports viewed by users who were/were not familiar with each Web page. We observe that users who are less familiar with the page view more viewports before perceiving a page to be loaded.

to be loaded in the 4G case. The results in the case of 3G, and even on the larger Pixel, are qualitatively similar. Taken together, our results show that users view an average of three viewports before deciding that the page is loaded, and do not wait for the initial viewport to load before scrolling.

Does familiarity affect viewing? During the user study, we include a survey to report if the user has visited the page or not. In our study, 50% of the pages are familiar to at least 35% of our users, and 5% of our pages are familiar to least 90% of our users. Figure 6 shows how the users mobile interaction changes based on their familiarity with the page for the 4G/Nexus 5 pairing. It is interpreted in the same manner as Figure 5. A user familiar with a page views only 1.8 viewports of the page before perceiving the page to be loaded, in the median case. In contrast, a user views a

median of 2.8 viewports on unfamiliar pages before perceiving the page to be loaded. In other words, users who are not familiar with pages tend to be more exploratory.

5.3 User scrolling behavior on desktop browsers

We conducted a crowdsourced study on desktop browsers with 50 users across the desktop variants of same 50 Web pages (§4). Our main finding, shown in Figure 5 is that the median user only looks at the initial viewport before deciding if a page is loaded (see label “Desktop”). This result corroborates with the wide spread focus on the first viewport for defining page load times in desktop/laptop browsers.

Our intuition for this result is that desktop screens are larger, and as a result more content is available to the user during the page load. Thus, users do not need to scroll to acquire more content during the transient process. In contrast, smaller mobile screens and more direct access to scrolling causes users to view multiple viewports before engaging with the page. Our studies with scrolling across platforms empirically confirm these points.

5.4 Targeted Task Study

We now analyze our other more task-based user study. The users perform these tasks on the same videos described in our earlier user study (§4). The tasks are distributed evenly throughout the viewports of the Web pages, some require the user to scroll and some do not. The time the user completed the task is taken as their perceived load. This user study was conducted across 50 users and pages, with videos recorded on the Nexus 5 on a 4G network.

Similar to the free browsing study, 70% of the users started scrolling before the first viewport was fully loaded, even when the task needed to be performed was in the initial viewport. We conclude from this that the users view multiple viewports even when they are not freely browsing.

6 MUPLT: MOBILE USER-CENTERED PLT

We define muPLT, a ground truth mobile user-defined PLT metric that we obtain from the user studies. We also show that the three traditional PLT metrics conventionally used to measure PLT—OnLoad [4], SpeedIndex [20], and Time to First Contentful Paint (FCP) [36] are significantly different compared to the muPLT measure.

6.1 Intuition behind the muPLT

muPLT is the time until the user perceives that the mobile page is loaded and they can start to engage with the page. We measure this implicitly based on the time between a user’s scrolls; the inter-scroll time should be small when the user is only viewing/searching the page, but should increase when the user is engaged with the content. Restated, a long pause in scrolling suggests that the user has likely stopped to consume the page contents and we use the long pause to measure ground truth.

A key question is: why do we capture muPLT this way? Recent works on desktops [21, 33, 45] have measured the user experience by explicitly asking users when they perceive the page to be loaded. However, this explicit feedback has high variability because of the

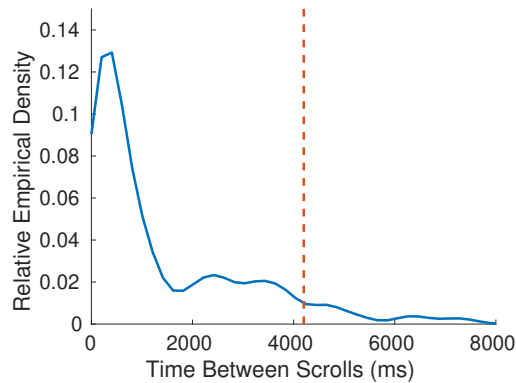


Figure 7: A distribution of the time between scrolls of users of theonion.com on a 4G network and Nexus5 phone. We mark user’s first inter-scroll time that is an outlier on this distribution as their muPLT.

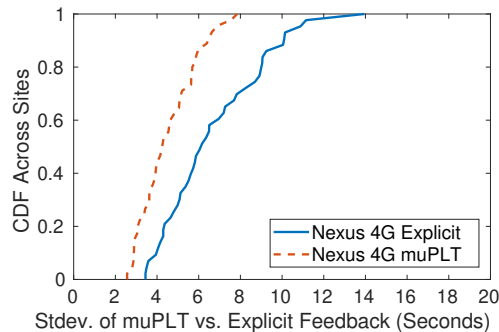


Figure 8: CDFs of the standard deviations across users for muPLT and explicit feedback. The CDFs shown is for one phone and network but the others show quantitatively similar results. The standard deviation of muPLT is much lower, suggesting the muPLT is more robust than the explicit feedback.

dual task paradigm [43]. This phenomenon states that a person’s ability to perform a task may degrade in the presence of having to performing another. In our study, the user is not only interacting with the page via scrolling, but also manually providing feedback on their perceived load time causing higher variability across users. As user agreement in crowdsourcing tasks is paramount [6], we wish to reduce the variability in our user-centered PLT measurements as much as possible.

We thus take the recommended approach by the the cognitive science community [43], which is to implicitly measure user responses. Implicit strategies for collecting user responses have been shown to result in much higher amounts of user agreement in the presence of the dual task.

6.2 Ground truth muPLT

We calculate muPLT for a user on a given page as the time in which a long pause occurs in that user’s scrolling behavior on that page. Specifically, we define the long pause as the first outlier in the empirical distribution of all users’ inter-scroll times. Figure 7 shows the empirical distribution of time between scrolls for *all* users over 4G network when perusing the load of theonion.com. The first point that is a statistical outlier (we use Tukey’s classical definition [44]) is our long pause. In Figure 7, a large majority of the scroll events (70%) are less than 2 seconds, where the first outlier on the distribution is at 4.2 seconds. We thus measure the muPLT of any user on theonion.com as the first time his or her inter-scroll time exceeds 4.2 seconds.

6.3 Comparing muPLT to explicit feedback

We design the muPLT metric based on user scrolling behavior, because the explicit feedback from the user shows high variance. Figure 8 shows the CDF of standard deviation across users when estimating muPLT using the users’ implicit scrolling behavior versus their explicit feedback. The results shown in Figure 8 are for the Nexus phone on the 4G network, but the results are similar for other combinations of phones and network conditions. Moreover, there is agreement between the muPLT metric and the explicit metric for user-centered PLT (Pearson correlation coefficient of .91 for 4G and .84 for 3G),

Finally, we find that on desktops, there is no need for an implicit metric because there is general agreement among users on when they perceive a page to be loaded. This is likely because the user is not interacting with the page before it is loaded, focusing on only one task, rather than the dual task. Our experiment (not shown here for brevity) shows that there is less than a 2 second standard deviation for the explicit feedback on desktops across the majority of the pages.

6.4 Comparing muPLT to PLT Metrics

We now compare the muPLT to the state-of-the-art PLT metrics. Figure 9 shows the percentage difference between muPLT and the three most popular PLT metrics—OnLoad, SpeedIndex, and FCP. For the median page, the difference between the user-centered metric and the traditional PLT metric is between 42% to 70% across network conditions. We conclude with these errors that existing systematic PLT metrics cannot estimate the user experience of page loads for mobiles.

We find that OnLoad still over- to under-estimates the page load performance due to it either waiting for too much content to be loaded, or missing dynamic content. The SpeedIndex and FCP, while visually oriented, only consider one viewport, whereas mobile users visit multiple viewports during the page load process (See Figure 1 for an example of these issues in practice).

7 DEVELOPING A MODEL FOR MUPLT_{EST}

Traditional metrics such as OnLoad, SpeedIndex, and FCP can be measured using automated tools during the page load process. As a result, a developer can scalably and programmatically obtain these metrics to measure page performance. However, these metrics do not match the user perception. User-centered metrics such as uPLT

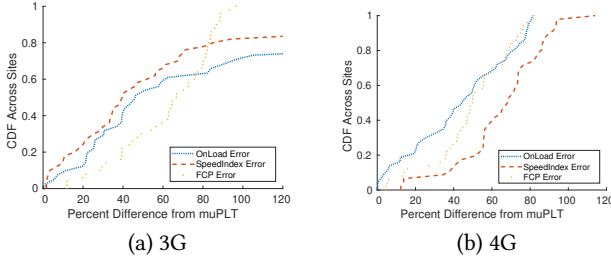


Figure 9: CDFs of the absolute percentage errors between muPLT and traditional PLT metrics—OnLoad, SpeedIndex, and FCP. Traditional metrics are not good measures of user experience as given by muPLT and see a 42% to 70% difference from muPLT.

(the explicit feedback used in previous studies) [21] and muPLT (described in this work) do capture the user defined load times on desktops and mobiles respectively, but they are reliant on large-scale user studies that do not scale well.

The primary objective of this work is model the mobile user-centered PLT (muPLT) using an estimator, muPLT_{est} . We design muPLT_{est} so that: (i) it can be scalably estimated using existing browser tools without requiring extensive user studies beyond an initial training phase, and (b) it accurately measures muPLT.

7.1 muPLT_{est} intuition

Our first intuition is that muPLT_{est} will depend on different aspects of the page load process, including the download times of objects and visual rendering of objects. Since mobile users scroll through multiple viewports before perceiving a page to be loaded, it is important to measure these aspects on multiple viewports.

The second intuition is that traditional PLT metrics do already measure different aspects of the page load. OnLoad measures the speed at which objects are loaded, SpeedIndex measures the visual rendering speed of the page, and FCP measures the first time a user perceives a visual difference in the page. Figure 10 shows the correlation between muPLT and each metric. Combining this information with Figure 9 shows that, while no metric captures muPLT, they do exhibit a useful linear relationship.

muPLT_{est} combines all of these metrics together, to jointly consider all these different aspects of the page load process. muPLT_{est} first estimates these metrics over multiple viewports and then combines them linearly. Further, these metrics are standardized across browsers (See Section 3) and are readily available to Web developers, thus, using only these metrics as inputs adds to the accessibility of the muPLT_{est} . By using existing metrics to predict muPLT, developers can easily estimate the user-centered latency without requiring extensive user studies.

7.2 Defining page load measures across multiple viewports

We first describe how we measure the visual progress of the page across multiple viewports using a new metric, which we call Multiple SpeedIndex. The traditional SpeedIndex metric is computed as

Metric	Absolute Error 3G	Absolute Error 4G
	<i>Mean ± SD seconds</i>	
OnLoad	21.96 ± 27.7	3.91 ± 5.33
SpeedIndex	6.32 ± 6.81	4.74 ± 3.45
FCP	6.63 ± 3.30	5.07 ± 1.06
muPLT_{est}	1.96 ± 1.61	0.727 ± 0.513

Table 1: Summaries of the performance of each PLT metric and the muPLT_{est} to the muPLT.

average visual completeness of the page. SpeedIndex measures the page load speed by looking at video frames of the page as it loads and computing the average time at which the pixels match the final state of the screen from the video. Traditionally, only a video of the first viewport is used. The formula for estimating SpeedIndex is:

$$\text{SpeedIndex} = \int_{t=0}^T 1 - VC(t) dt, \quad (1)$$

Where VC is the visual completeness of a given viewport at time t . We compute Multiple SpeedIndex as:

$$\text{Multiple SpeedIndex} = \sum_{i=1}^n \alpha_i * \text{SpeedIndex}_i \quad (2)$$

$$s.t. \sum_{i=1}^n \alpha_i = 1 \quad (3)$$

Multiple SpeedIndex is estimated using a simple weighted average of the SpeedIndex over multiple viewports, thereby adding visual information from multiple parts of the page to the estimator. For our calculation of Multiple SpeedIndex, we simply set the α values to $1/n$ where n is the number of videos shown to the users for the page. This new metric has a correlation coefficient of 0.63 with the muPLT metric on 4G; in contrast SpeedIndex estimated over the initial viewport only has a correlation coefficient of 0.21.

Similarly, for the FCP metric, we estimate the weighted average of FCP over multiple viewports. The OnLoad metric is the same for all viewports because it is estimated as the total time to load all objects, which we have shown does not change across each viewport (Section 4).

7.3 Estimating muPLT_{est}

We use a LASSO regression [42] to combine the new metrics estimated over multiple viewports to model muPLT_{est} . Given the nature of the scatterplot in Figure 10, we assume that a linear relationship between muPLT and each metric will suffice. With this, we can formally describe our model as:

$$\text{muPLT}_{est} = w^T \text{PLT} + \epsilon \quad (4)$$

where PLT represents a matrix with a row for each page, and columns for the OnLoad, Multiple SpeedIndex, SpeedIndex, and FCP of each page. The choice of weights, w , and bias b , on the PLT metrics are learned by minimizing the error between the regularized linear sum of the metrics, or:

$$\underset{w, b}{\text{minimize}} \lambda \|w\| + \sum_{i=1}^n (w^T \text{PLT}_i + b - \text{muPLT}_i)^2 \quad (5)$$

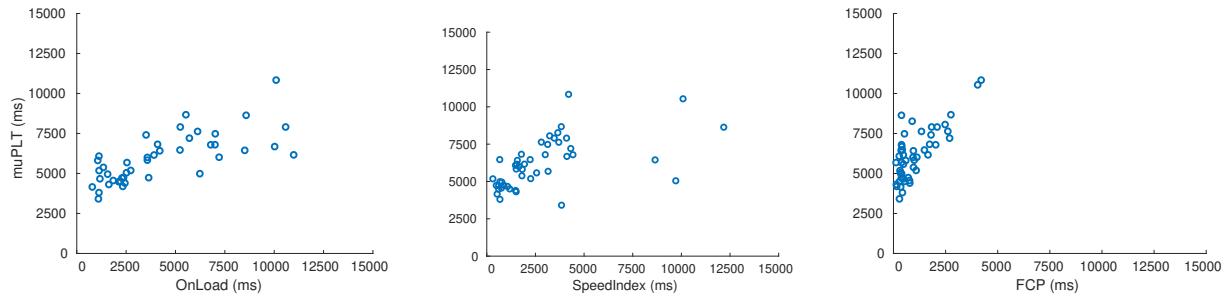


Figure 10: Scatter plots of each page load metric against the muPLT on a 4G network. Each time is given in milliseconds. There is roughly a linear relationship, highlighting our choice of a linear regression model used for the muPLT_{est} .

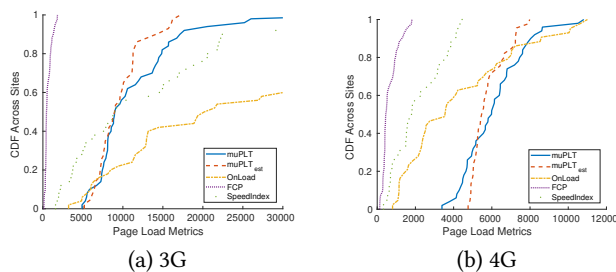


Figure 11: CDFs of muPLT and muPLT_{est} . muPLT_{est} is close in distribution to muPLT, with at most 15% error across networks at the median. Compared to traditional PLT metrics (OnLoad, SpeedIndex, FCP) muPLT_{est} provides a significantly better predictor to the user experience.

In order to choose the best model, we tune the LASSO for 100 values of the λ parameter and choose the result that gives the least Mean Square Error (MSE). This common parameter tuning procedure also performs variable selection over our PLT inputs. To avoid over fitting, the MSE at each iteration is defined using a leave-out-5, 10 fold cross validation. The resulting model gives us a prediction muPLT_{est} for each page.

Of course, model training and obtaining the weights requires ground truth muPLT values from user studies. But once trained, muPLT_{est} can be estimated without further user studies. In fact, in the next section we show the muPLT_{est} can be used to accurately estimate muPLT even when the Web pages change due to optimizations.

7.4 Evaluating muPLT_{est}

We evaluate the effectiveness of muPLT_{est} in terms of how well it estimates the ground truth user-centered page load across the 50 Web pages. Figure 11 shows the CDFs of muPLT_{est} and muPLT across all the Web pages and users. The performance of each metric in predicting the muPLT is summarized via Table 1 which gives the means and standard deviations for each of these CDFs. In addition, the performance of the muPLT_{est} in terms of the Root Mean Square Error (RMSE) of the predictions was .934 and 2.53 for 3G and 4G respectively. As to the composition of the model, all weights were

non-zero, with the Multiple SpeedIndex contributing about $7\times$ more than the SpeedIndex to the final prediction. Our key takeaway is that muPLT_{est} is a good predictor of muPLT, with a median error of 15% in the 3G network and 10% in the 4G network.

Recall that the median error between each of the traditional PLT metrics and muPLT was between $\approx 44\text{--}90\%$ depending on the metric (see Figure 9). In terms of the RMSE, these other metrics also perform poorly, with the best case being 5.19 and 7.53 for 3G and 4G over the OnLoad, SpeedIndex, and FCP. Further, even models trained using these metrics individually were only able to perform with a max RMSE of 4.53 across networks, suggesting the power of combining each metric in predicting the final muPLT. We use a 10 fold, leave-5-out cross validation as our estimate of the predictive power of the model.

The correlation between muPLT_{est} and muPLT is 0.92 and 0.84 for 3G and 4G respectively. Importantly, muPLT_{est} can be estimated in real time as the page loads and does not require extensive user studies, except for the initial training.

We conclude our evaluation of the muPLT_{est} by discussing factors which allow the model to generalize. Firstly, the RMSE of a more general model for the muPLT_{est} which included predictions on 3G and 4G data was 2.21, suggesting the ability of the model to scale across networks. Though the majority of experiments and data were taken across Google platforms, the PLT metrics used to train the muPLT_{est} are all standardized across modern browsers (See Section 3). All such browsers implement scrolling similarly [25], alluding to the similarity in ground truth for the model on different platforms. Further, we provide an illustrative example in the following section for the ability of the model to make predictions under a different network protocol.

8 USING THE USER-CENTERED MODEL IN PRACTICE

We provide an application of muPLT_{est} in terms of how it can help developers choose the right Web page optimization to improve the muPLT of their page. Many optimizations have been designed to improve page performance [30], but these optimizations have been designed to improve traditional PLT metrics. Their effect on muPLT has not been a first class concern. Further, the effect of the optimization often depends on the choice of parameters or requires heavy configuration.

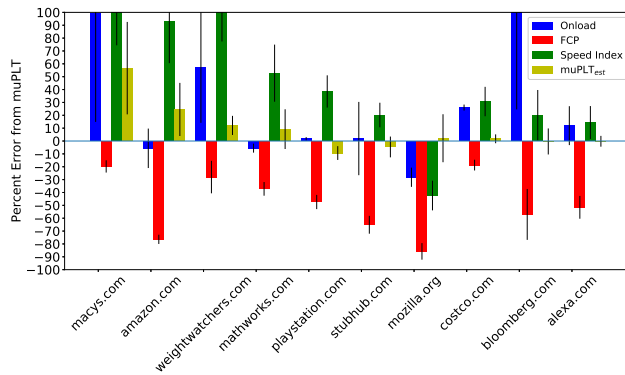


Figure 12: Evaluating how well muPLT_{est} estimates the user-centered load time (muPLT) when different ServerPush optimization strategies are applied to Web pages. The bars show the prediction errors of muPLT_{est} , OnLoad, FCP, and SpeedIndex when compared to ground truth muPLT. muPLT_{est} differs from the ground truth by a median 7.8% across pages. Other metrics have much larger errors.

Without the muPLT_{est} model, developers will have to launch large scale user studies, using platforms such as EyeOrg [45], to get user-centered PLT for each possible configuration of the optimization. This clearly does not scale. Instead, muPLT_{est} provides a scalable model to compare optimizations with respect to user experience. We now show how a developer can use muPLT_{est} to evaluate the effect of one optimization—ServerPush [30]—under different configurations.

8.1 Application case study: ServerPush

Server Push allows servers that host Web pages to preemptively *push* objects needed to load the page to the client’s Web browser. ServerPush is notoriously hard to get right because the right class of objects to push depends on the Web page structure and available bandwidth [9, 47]. Getting ServerPush right can optimize the page load, but getting it wrong can slow the page down. For example, in some Web pages, pushing all the JavaScripts (JS) on the Web page is beneficial because the JS may embed further objects within the page. But, in other Web pages, pushing Cascading Style Sheets (CSS) may be more important, especially for those where rendering is complex. Further, pushing too many objects can hurt performance because they all compete for limited bandwidth.

We use the muPLT_{est} model to predict the user-centered latency when using ServerPush under the following strategies: (1) default: no ServerPush, (2) push *js*: the server pushes all JS objects, (3) push *css*: the server pushes all CSS, and (4) push *css + js*: The server pushes all JS and CSS on the page. We implement each strategy using our own HTTP Record and Replay [28, 34] server written in NodeJS.

To test the muPLT_{est} under the different strategies we randomly selected 10 Web pages from our previous 50 pages. For each Web page, we implement the 4 alternative ServerPush strategies discussed. We then use muPLT_{est} to predict the user-centered PLT for each strategy, and report the results.

To verify that our estimates are correct, we collect ground truth muPLT_{est} values using a user study with 50 crowdsourced users over the Microworker platform. We use the same methodology as previous described in Section 4. We note that this user study is only to verify that our muPLT_{est} estimates are accurate under ServerPush configurations; a developer will not have to run this user study.

8.2 Evaluating muPLT_{est} with ServerPush strategies

We first verify that ServerPush does indeed make a difference in user-defined delays for page loads. Thus, a well fit user-centered model should capture these changes.

Figure 12 shows the mean error of different metrics with respect to the muPLT ground truth across all alternative strategies for the 10 pages. The height of each bar shows the error, in terms of the standard deviation across the versions, for a given page.

muPLT_{est} provides a reliable way to estimate the improvement in user perception for different Web optimizations. The metric exhibits a median absolute error to the muPLT of 7.8% across pages, with the next closest being the OnLoad which exhibits a 23% median absolute error, and a much larger error variation between the pages. Finally all other metrics are biased in their errors, they either over or under predict drastically. By contrast the errors of the muPLT_{est} are much more centered about zero, speaking to the model’s validity.

In conclusion, using muPLT_{est} , a developer can pick the right ServerPush strategy to specifically improve a key user experience metric for their Web pages without requiring extensive user studies. In practice, developers can make use of muPLT_{est} in the same manner to evaluate the effect of other optimizations on their pages.

9 CONCLUSIONS

We developed a complete methodology to model and estimate user-centered Page Load Time (PLT) of Web pages on smartphones. To do so, we created a platform for standardized crowdsourced measurements of the user’s perception of mobile page load time. The key difference between mobile and desktop browsing in terms of user-interactions is that mobile users view multiple viewports before even before the page content is loaded; in contrast, desktop users only view the initial viewport until the page is loaded. The crowdsourced user study we conduct allows scrolling behavior to play a part in capturing a user’s perceived delay. We draw various conclusions from this study, including that more than just the first viewport of content matters to mobile users. Based on the user study, we define an implicit perceptual metric called muPLT , which stands for mobile user-centered PLT. We find that muPLT differs substantially from traditional PLT metrics. Although muPLT captures user perceived delay, it requires large scale user studies to measure. Instead, we design an estimator muPLT_{est} that accurately captures muPLT . We show that our estimator can predict muPLT to within 10-15% error at the median across 50 pages. Finally, we show how developers and researchers can use the muPLT_{est} to estimate the user experience when applying different Web page optimizations without the need for large scale user studies.

REFERENCES

- [1] [n. d.]. Microworkers: Crowdsourcing platform. <https://microworkers.com/>.
- [2] [n. d.]. Mobile vs Desktop: 10 Key differences. <https://www.paradoxlabs.com/blog/mobile-vs-desktop-10-key-differences/>.
- [3] [n. d.]. Moving beyond window.onload(). <https://bit.ly/2Mgh3OJ>.
- [4] [n. d.]. The onload property of the GlobalEventHandlers. <https://mzl.la/1GnvIMe>.
- [5] Alexa. [n. d.]. Top Sites in the United States. <http://bit.ly/2ybNLcb>.
- [6] Abdullah X. Ali, Meredith Ringel Morris, and Jacob O. Wobbrock. 2018. Crowdsourcing Similarity Judgments for Agreement Analysis in End-User Elicitation Studies. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology (UIST '18)*. ACM, New York, NY, USA, 177–188.
- [7] Kayce Basques. [n. d.]. Analyzing Network Performance in Chrome DevTools. <http://bit.ly/2tzvjpj>.
- [8] Frank R. Bentley, S. Tejaswi Peesapati, and Karen Church. 2016. "I Thought She Would Like to Read It": Exploring Sharing Behaviors in the Context of Declining Mobile Web Use. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 1893–1903.
- [9] Tom Bergan, Simon Pelchat, and Michael Buettner. [n. d.]. Rules of Thumb for HTTP/2 Server Push. <https://bit.ly/2Wwagll>.
- [10] Michael S. Bernstein, Greg Little, Robert C. Miller, Björn Hartmann, Mark S. Ackerman, David R. Karger, David Crowell, and Katrina Panovich. 2010. Soylent: A Word Processor with a Crowd Inside. In *Proceedings of the 23rd Annual ACM Symposium on User Interface Software and Technology (UIST '10)*. ACM, New York, NY, USA, 313–322.
- [11] Joshua Bixby. 2012. 4 awesome slides showing how page speed correlates to business metrics at Walmart .com. <https://bit.ly/2kILgdu>.
- [12] Juan Pablo Carrascal and Karen Church. 2015. An In-Situ Study of Mobile App & Mobile Search Interactions. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 2739–2748.
- [13] Chrome Debug Team. [n. d.]. Chrome Remote Debugging. <http://bit.ly/2rnmsZx>.
- [14] Yanqing Cui and Virpi Roto. 2008. How People Use the Web on Mobile Devices. In *Proceedings of the 17th International Conference on World Wide Web (WWW '08)*. ACM, New York, NY, USA, 905–914.
- [15] Biplab Deka, Zifeng Huang, Chad Franzen, Jeffrey Nichols, Yang Li, and Ranjitha Kumar. 2017. ZIPT: Zero-Integration Performance Testing of Mobile App Designs. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST '17)*. ACM, New York, NY, USA, 727–736.
- [16] Kit Eaton. [n. d.]. How One Second Could Cost Amazon 1.6 Billion In Sales. <http://bit.ly/1Beu9Ah>.
- [17] Leah Findlater, Jon E. Froehlich, Kays Fattal, Jacob O. Wobbrock, and Tanya Dast-yar. 2013. Age-related Differences in Performance with Touchscreens Compared to Traditional Mouse Input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 343–346.
- [18] Leah Findlater, Joan Zhang, Jon E. Froehlich, and Karyn Moffatt. 2017. Differences in Crowdsourced vs. Lab-based Mobile and Desktop Input Performance Data. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 6813–6824. <https://doi.org/10.1145/3025453.3025820>
- [19] R. Geng and J. Tian. 2015. Improving Web Navigation Usability by Comparing Actual and Anticipated Usage. *IEEE Transactions on Human-Machine Systems* 45, 1 (Feb 2015), 84–94. <https://doi.org/10.1109/THMS.2014.2363125>
- [20] Daniel Imms. 2014. Speed Index: Measuring Page Load Time a Different Way. <http://bit.ly/2dbuUpr>.
- [21] Conor Kelton, Jihoon Ryou, Aruna Balasubramanian, and Samir R. Das. 2017. Improving User Perceived Page Load Times Using Gaze (NSDI '17). USENIX Association, 545–559. <https://www.usenix.org/conference/nsdi17/technical-sessions/presentation/kelton>
- [22] Aniket Kittur, Boris Smus, Susheel Khamkar, and Robert E. Kraut. 2011. CrowdForge: Crowdsourcing Complex Work. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*. ACM, New York, NY, USA, 43–52.
- [23] Steven Komarov, Katharina Reinecke, and Krzysztof Z. Gajos. 2013. Crowdsourcing Performance Evaluations of User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 207–216. <https://doi.org/10.1145/2470654.2470684>
- [24] Walter S. Lasecki, Rachel Wesley, Jeffrey Nichols, Anand Kulkarni, James F. Allen, and Jeffrey P. Bigham. 2013. Chorus: A Crowd-powered Conversational Assistant. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology (UIST '13)*. ACM, New York, NY, USA, 151–162.
- [25] Nolan Lawson. 2017. Scrolling on the web: A primer. <https://blogs.windows.com/msedgedev/2017/03/08/scrolling-on-the-web/>.
- [26] Byungjoo Lee, Olli Savisaari, and Antti Oulasvirta. 2016. Spotlights: Attention-Optimized Highlights for Skim Reading. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 5203–5214. <https://doi.org/10.1145/2858036.2858299>
- [27] Inseong Lee, Jaesoo Kim, and Jinwoo Kim. 2005. Use Contexts for the Mobile Internet: A Longitudinal Study Monitoring Actual Use of Mobile Internet Services. *International Journal of Human Computer Interaction* 18, 3 (2005), 269–292. https://doi.org/10.1207/s15327590ijhc1803_2
- [28] Helen Li. [n. d.]. Web Page Replay Go. <http://bit.ly/2p9fUe4>.
- [29] Sarah Lim, Joshua Hibschan, Haoqi Zhang, and Eleanor O'Rourke. 2018. Ply: A Visual Web Inspector for Learning from Professional Webpages. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology (UIST '18)*. ACM, New York, NY, USA, 991–1002. <https://doi.org/10.1145/3242587.3242660>
- [30] M. Thomson M. Belshe, R. Peon. [n. d.]. Hypertext Transfer Protocol Version 2 (HTTP 2). <https://httpwg.org/specs/rfc7540.html>.
- [31] Raphael Menges, Hanadi Tamimi, Chandan Kumar, Tina Walber, Christoph Schaefer, and Steffen Staab. 2018. Enhanced Representation of Web Pages for Usability Analysis with Eye Tracking. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications (ETRA '18)*. ACM, New York, NY, USA, Article 18, 9 pages. <https://doi.org/10.1145/3204493.3204535>
- [32] Michael Nebeling and Anind K. Dey. 2016. XDBrowser: User-Defined Cross-Device Web Page Designs. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 5494–5505.
- [33] Ravi Netravali, Vikram Nathan, James Mickens, and Hari Balakrishnan. 2018. Vesper: Measuring Time-to-Interactivity for Web Pages. In *15th USENIX Symposium on Networked Systems Design and Implementation (NSDI '18)*. USENIX Association, Renton, WA, 217–231. <https://www.usenix.org/conference/nsdi18/presentation/netravali-vesper>
- [34] Ravi Netravali, Anirudh Sivaraman, Somak Das, Ameesh Goyal, Keith Winstein, James Mickens, and Hari Balakrishnan. 2015. Mahimahi: Accurate Record-and-Replay for HTTP. In *2015 USENIX Annual Technical Conference (USENIX ATC 15)*. USENIX Association, Santa Clara, CA, 417–429. <https://www.usenix.org/conference/atc15/technical-session/presentation/netravali>
- [35] Dan Oksnevad. [n. d.]. Time to Interact: A New Metric for Measuring User Experience. <http://bit.ly/2Fz0mvd>.
- [36] Shubhie Panicker. [n. d.]. W3C Paint Timing Working Draft. <http://bit.ly/2f2CGSk>.
- [37] Fabio Paternò, Antonio Giovanni Schiavone, and Antonio Conti. 2017. Customizable Automatic Detection of Bad Usability Smells in Mobile Accessed Web Applications. In *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '17)*. ACM, New York, NY, USA, Article 42, 11 pages. <https://doi.org/10.1145/3098279.3098558>
- [38] Joel Ross, Lilly Irani, M. Six Silberman, Andrew Zaldivar, and Bill Tomlinson. 2010. Who Are the Crowdworkers?: Shifting Demographics in Mechanical Turk. In *CHI '10 Extended Abstracts on Human Factors in Computing Systems (CHI EA '10)*. ACM, New York, NY, USA, 2863–2872. <https://doi.org/10.1145/1753846.1753873>
- [39] Sujan Shrestha. 2007. Mobile Web Browsing: Usability Study. In *Proceedings of the 4th International Conference on Mobile Technology, Applications, and Systems and the 1st International Symposium on Computer Human Interaction in Mobile Technology (Mobility '07)*. ACM, New York, NY, USA, 187–194.
- [40] Android Studio. [n. d.]. The Android Debug Bridge (adb). <http://bit.ly/2jOqfuz>.
- [41] Kazuki Takashima, Nana Shinshi, and Yoshifumi Kitamura. 2015. Exploring Boundless Scroll by Extending Motor Space. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '15)*. ACM, New York, NY, USA, 557–566. <https://doi.org/10.1145/2785830.2785884>
- [42] Robert Tibshirani. 1996. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)* 58, 1 (1996), 267–288. <http://www.jstor.org/stable/2346178>
- [43] Naotsugu Tsuchiya and Christof Koch. 2016. The Relationship Between Consciousness and Top-Down Attention. <https://www.sciencedirect.com/science/article/pii/B9780128009482000054>. In *The Neurology of Consciousness (Second Edition)*. 71–91.
- [44] John W. Tukey. 1977. Exploratory Data Analysis. Addison-Wesley, 545–559.
- [45] Matteo Varvello, Jeremy Blackburn, David Naylor, and Konstantina Papagiannaki. 2016. EYEORG: A Platform For Crowdsourcing Web Quality Of Experience Measurements (CoNEXT '16). ACM, 399–412. <https://doi.org/10.1145/2999572.2999590>
- [46] P. Wang, M. Varvello, and A. Kuzmanovic. 2019. Kaleidoscope: A Crowdsourcing Testing Tool for Web Quality of Experience. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. 1971–1982.
- [47] Torsten Zimmermann, Benedikt Wolters, and Oliver Hohlfeld. 2017. A QoE Perspective on HTTP/2 Server Push. In *Proceedings of the Workshop on QoE-based Analysis and Management of Data Communication Networks (Internet QoE '17)*. ACM, New York, NY, USA, 1–6.
- [48] Torsten Zimmermann, Benedikt Wolters, Oliver Hohlfeld, and Klaus Wehrle. 2018. Is the Web Ready for HTTP/2 Server Push?. In *Proceedings of the 14th International Conference on Emerging Networking EXperiments and Technologies (CoNEXT '18)*. ACM, New York, NY, USA, 13–19.