On Landing and Internal Web Pages

The Strange Case of Jekyll and Hyde in Web Performance Measurement

Waqar Aqeel
Duke University & M.I.T.

Balakrishnan Chandrasekaran
Max-Planck-Institut für Informatik

Anja Feldmann
Max-Planck-Institut für Informatik

Bruce M. Maggs
Duke University, Emerald Innovations, & M.I.T.

ABSTRACT

There is a rich body of literature on measuring and optimizing nearly every aspect of the web, including characterizing the structure and content of web pages, devising new techniques to load pages quickly, and evaluating such techniques. Virtually all of this prior work used a single page, namely the landing page (i.e., root document, "/"), of each web site as the representative of all pages on that site. In this paper, we characterize the differences between landing and internal (i.e., non-root) pages of 1000 web sites to demonstrate that the structure and content of internal pages differ substantially from those of landing pages, as well as from one another. We review more than a hundred studies published at top-tier networking conferences between 2015 and 2019, and highlight how, in light of these differences, the insights and claims of nearly two-thirds of the relevant studies would need to be revised for them to apply to internal pages.

Going forward, we urge the networking community to include internal pages for measuring and optimizing the web. This recommendation, however, poses a non-trivial challenge: How do we select a set of representative internal web pages from a web site? To address the challenge, we have developed *Hispar*, a "top list" of 100,000 pages updated weekly comprising both the landing pages and internal pages of around 2000 web sites. We make *Hispar* and the tools to recreate or customize it publicly available.

CCS CONCEPTS

• **Networks** → **Network measurement**; • **Information systems** → **World Wide Web**;

KEYWORDS

Web page performance, PLT, Speed Index, QoE, top lists

ACM Reference Format:
Waqar Aqeel, Balakrishnan Chandrasekaran, Anja Feldmann, and Bruce M. Maggs. 2020. On Landing and Internal Web Pages: *The Strange Case of Jekyll and Hyde in Web Performance Measurement*. In *ACM Internet Measurement Conference (IMC '20), October 27–29, 2020, Virtual Event, USA.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3419394.3423626

IMC '20, October 27–29, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-8138-3/20/10…$15.00
https://doi.org/10.1145/3419394.3423626

1 INTRODUCTION

Any attempt to quantify a characteristic of a web site raises the following question: What page or pages of the site should be used for the quantification? A cursory review of a decade's worth of literature on web performance measurement and optimization (abbreviated, henceforth, as *web-perf.*) reveals that, until now, the implicit answer to that question has been the landing page. The landing page (i.e., root document, "/") of a web site is quite important. It serves as the primary gateway through which users discover content on the site. But internal pages (i.e., non-root documents) are often equally important. For example, the content consumed by users (e.g., articles on news web sites and posts from friends on social media platforms) are typically served on internal pages. This importance is also reflected, for instance, in the attention paid to internal pages in search engine optimization, which helps publishers in monetizing their content by driving traffic to their web sites from search engines [41]. Why, then, is it the case that almost all prior web-perf. studies ignore the internal pages and focus only on the landing pages of web sites?

Prior work implicitly assumes that the performance measures and optimizations of landing pages generalize to most, if not all, internal pages. We use the term "web performance study," to refer loosely to a broad range of efforts: characterizing one or more aspects of web pages (e.g., distribution of different types of objects, prevalence of ads and trackers, and adoption of specific security features), estimating and improving the load and display times of pages, and evaluating novel optimizations to reduce the page-load times. To measure or optimize web page performance, studies typically use one or more rank-ordered lists or *top lists* of web sites, e.g., Alexa [6] and Quantcast [87]. The top lists provide only the domain name of a web site, such as `nytimes.com` or `www.wikipedia.org`. After choosing a web site from a top list, researchers typically use the landing page of that site in their experiments. Every aspect of these web-perf. studies—metrics, optimizations, evaluation techniques, and even characteristics of top lists—has faced extensive scrutiny [67, 84, 93], except one: the exclusion of internal pages.

The exclusion of internal pages might have been intentional. The rationale might be that the differences between landing and internal pages, if any, are random—a simple statistical problem remedied by measuring a large number of landing pages. It may also be that the page-type differences are common knowledge and the studies are page-type agnostic. This paper casts doubt on both rationales.

We compare the landing page of a web site with several internal pages of that site and repeat the analyses for 20,000 pages from around 1000 web sites. We show that internal pages differ substantially in content and performance from landing pages; internal

pages also vary significantly from one another. The differences between the two page types also vary based on the popularity rankings of web sites. We manually review more than a hundred web-perf. studies published at top-tier networking venues and demonstrate that a significant fraction of them are affected by the exclusion of internal pages: to apply to internal pages, more than two-thirds of the relevant studies would have to revise their claims to avoid over-generalized insights or assertions. Hence we urge that all future web-perf. work analyze both landing pages and internal pages.

While there is no ambiguity in choosing landing pages, since there is only one per web site, the recommendation that all web-perf. studies should include internal pages poses a non-trivial challenge: How can we select a set of "representative" pages from the available internal pages of a web site? To address this challenge, we exploit the key objective behind web perf. studies—improving users' browsing experience. Given this intent, it is only logical to select internal pages visited by real users. Thus, we use search engines to find "popular," or frequently visited internal pages of web sites; we assume that these pages are representative of the typical internal pages that users visit at these sites. To this end, we created *Hispar* ($\mathcal{H}$), a new top list that includes landing as well as internal pages of different web sites. Unlike current top lists, which provide only the domain names of web sites, *Hispar* comprises complete URLs of both landing pages and a subset of internal web pages. We use *Hispar* to characterize the differences between landing and internal pages of web sites and ascertain their impact on prior work.

We summarize our contributions as follows.

⋆ We create *Hispar* with around 1000 highest-ranked web sites ($\mathcal{H}_{1K}$) from the Alexa Top 1M, selecting for each the landing page and at most 19 frequently visited internal pages. *Hispar* uses search engine results for discovering internal pages. Our experiments against $\mathcal{H}_{1K}$ reveal that internal pages of a web site not only differ substantially from the landing page, but also from one another. We release our data set for use by other researchers [50].

⋆ We describe the page-type differences in detail and highlight the implications of each for prior work. For the latter, we review more than a hundred web-perf. measurement and optimization studies published at five premier networking conferences over the past five years, from 2015 to 2019, and show that two-thirds of the relevant publications would require some revision for the results to apply to internal pages.

⋆ We expand $\mathcal{H}_{1K}$ to generate a much larger list, $\mathcal{H}_{2K}$, that includes about 2000 web sites, with one landing and at most 49 internal web page URLs for each web site. We release $\mathcal{H}_{2K}$ and the tools for recreating or customizing *Hispar* as open source artifacts [50]. We discuss the stability of *Hispar*, present the economic feasibility of our approach, and outline alternative approaches for creating the list.

We hope that our findings and recommendations serve as a "call to arms" to the networking community to include internal pages when measuring and optimizing the web.

2 IMPACT ON PREVIOUS STUDIES

We conducted a brief survey of research on web performance measurement and optimization published at top-tier conferences and focused on answering two questions: *(a) How prevalent is the use of internal pages in such prior studies? (b) For studies that focus only on landing pages, would the inclusion of internal pages impact their claims or insights?*

Figure 1: *Nearly two-thirds of the web-perf. studies that use a top list and were published between 2015 and 2019 at 5 top-tier networking venues would require some revision for them to apply to internal pages.*

Venue	*Pubs.*	*#using top list*	*Revision Score*		
			Maj.	*Min.*	*No*
IMC	214	56	9	23	24
PAM	117	27	7	10	10
NSDI	222	11	6	4	1
SIGCOMM	187	9	1	6	2
CoNEXT	180	16	7	5	4

We reviewed papers published from 2015 to 2019 at five premier networking venues, namely *ACM Internet Measurement Conference (IMC)*, *Passive and Active Measurement Conference (PAM)*, *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, *ACM Special Interest Group on Data Communications (SIGCOMM)*, and *ACM Conference on emerging Networking EXperiments and Technologies (CoNEXT)*. We collected 920 papers in total and programmatically searched the PDF-versions of these papers for terms related to the five widely used top lists in the literature, viz., *Alexa* [5], *Majestic* [69], *Umbrella* [100], *Quantcast* [88], and *Tranco* [84]. We then manually inspected the papers with one or more matching terms to weed out false positives, e.g., papers that mention "Alexa" Echo Dot and have nothing to do with the "Alexa" top list, and those that mention a top list only when discussing prior work. After eliminating these false positives, we were left with 119 papers that used at least one of the top lists.

We manually reviewed each of these 119 papers to determine whether they used internal pages. We found that only 15 (12.6%) of the papers implicitly or explicitly use internal pages in their experiments. Seven papers, for example, analyzed web-browsing *traces* of users, and we assume that the URLs in these traces include both landing and internal pages of different web sites. Another set of eight papers, involving active measurements, took measures to include internal pages, either by recursively crawling a web site or *monkey testing* (e.g., randomly clicking buttons and links, and typing text to trigger navigation). The remaining 104 (87.4%) papers ignored internal pages in their studies.

We evaluated the remaining 104 papers to ascertain how their claims and insights might change had they included internal pages in their experiments. We captured the extent of this change via a *revision score* (refer Tab. 1) that takes one of three values, viz., *No revision*, *Minor revision*, and *Major revision*, on an ordinal scale.

No revision implies that the differences between page types are irrelevant for the study. We assign this label to a study if, for example, it is a trace-based study and uses a top list only to rank the web sites in the trace [9] or uses landing pages from a top list, but mixes in data from other sources to compose their data set [42].

Minor revision implies that although a given study uses a top list, its insights are not based solely on landing pages. Berger et al. [14], for example, uses landing pages to evaluate their system,

but they also conduct three other types of evaluation that are independent of or agnostic to page types. Similarly, one of the evaluation methods in [94] uses only landing pages from a top list to measure the performance overhead of their system.

Major revision implies that a given research work focuses chiefly on web page performance but excludes internal pages, or uses only landing pages to evaluate their proposals. Netravali et al. [76], for instance, proposes a web page delivery optimization and uses only landing pages to measure the improvement in page-load times brought about by the optimization. Bashir et al. [12] report on how tracking companies use WebSockets for circumventing ad blockers, but measure the prevalence of this practice only on landing pages.

From the 119 publications we review, we label 41 (34.5%) papers as requiring no revision, 48 (40.3%) as requiring a minor revision, and 30 (25.2%) papers as requiring a major revision. In short, the claims and insights of nearly two-thirds of all web-perf. publications published in the last five years in these five venues would need at least a minor revision in order to apply to internal pages. We also find that most papers do not comment on whether results derived from the analysis of landing pages also apply to internal pages.

Caveats. Our survey is limited to five venues, but these are top-tier conferences with a high bar for research quality. Although the majority of web-perf. studies published at these venues ignored internal pages in their experiments, it is hard to generalize our findings to other venues without further investigation. Lastly, while the revision scores are coarse and subjective, they are instructive for understanding the ramifications of this work.

3 THE *HISPAR* TOP LIST

To determine whether landing and internal pages of web sites differ significantly, ascertain how they differ, and investigate any implications for web performance measurement and optimization, we created a new top list, called *Hispar* ($\mathcal{H}$). Unlike typical top lists, *Hispar* consists of a list of URL sets, one for each web site. The URL set for a site comprises the landing page as well as a subset of the site's internal pages.

We bootstrap *Hispar* from a top-ranked subset of the Alexa Top 1 million [5] list ($\mathcal{A}_{1M}$) by replacing each domain in the latter with the landing page as well as a set of internal pages from the corresponding web site. While obtaining the landing-page URL for each web site in $\mathcal{A}_{1M}$ is straightforward, retrieving the internal-pages' URLs introduces several challenges. It is infeasible to exhaustively crawl all internal pages of all web sites. Performing an exhaustive crawl even on a small scale may be unethical. It may introduce fake page visits or ad impressions, distort the statistics that the web site collects, increase the load on the web server, and cost the web site money by creating bandwidth costs. We limit, hence, the set of internal pages per web site to at most N pages.

In the absence of a content provider's support in selecting a set of representative internal pages from their web site, we turn to search engines. Specifically, we use the Google Search Engine API [47] to discover a set of at most N internal pages for each web site. We opted for search-engine results as they are biased towards what people search for and click on [70]. The measurements conducted and optimizations tested on such internal pages are, hence, likely to reflect and improve the browsing experience of real users. Prior work has used search-engine queries with satisfactory results [78], and this method allows us to avoid exhaustive crawls.

Starting with the most popular site listed in $\mathcal{A}_{1M}$, we examine the sites one-by-one until *Hispar* has enough pages. For each web site ω, we used the Google Search Engine API for the term "`site:`ω." We fix the user's location for the search queries to the United States, and limit the search results to pages in the English language. We restrict our searches to web page URLs and filter out files such as PDF and Word documents. We drop any ω for which there are fewer than 10 results, which is typically the case with international web sites that have very few pages in English. We collect the top N unique web-page URLs (including the landing page) from the search results for each ω.

Using the above methodology, we generated *Hispar* containing 100,000 web page URLs. Referred to as $\mathcal{H}_{2K}$, the list contains at least 2000 URL sets of size $N = 50$, one for each web site. Each URL set contains one landing and at most 49 internal pages. We refresh $\mathcal{H}_{2K}$ once every week, and make the lists publicly available for researchers. The size and refresh rate are limited to reduce the cost of publishing this (free) list. We release the tools and artifacts required for regenerating or customizing the lists [50].

We think the list size is sufficient as 93% of studies that received a major revision score in our survey (in §2), i.e. studies that would benefit most from this list, measured 100,000 or fewer pages. We refresh $\mathcal{H}_{2K}$ every Thursday at 11AM UTC. We avoid weekends, since weekly patterns in Internet traffic affect the $\mathcal{A}_{1M}$ list [93], which we use for bootstrapping $\mathcal{H}_{2K}$. We also do not randomize the day of the week to keep the frequency of updates static.

Why "Alexa" and not others? The choice of using the Alexa top list to bootstrap *Hispar* is somewhat arbitrary. Alternatives include Cisco Umbrella [100], Majestic million [69], Quantcast [88], and Tranco [84]. Cisco Umbrella ranks web sites based on the volume of DNS queries issued for the domains as well as the number of unique client IP addresses requesting a domain's resolution [54]. The fully qualified domain names (FQDNs) in the list, as a consequence, do not necessarily reflect end-user browsing behavior: on `2019-06-15`, 4 of the top 5 entries, for instance, were Netflix domains. Majestic ranks web sites by the number of unique IP subnets hosting servers that serve pages containing links to a domain, which is more of a measure of quality than traffic [68]. Tranco combines four lists including Umbrella and Majestic [84], which we did not want to use. Between Alexa and Quantcast, we chose Alexa because of its popularity: only 10 (or 8.4%) out of 119 papers in our survey (§2) use a top list other than Alexa. The justification notwithstanding, our study is agnostic to which top list is used for bootstrapping *Hispar*, since none of the top lists include internal pages.

Why use search engine results? Alternative ways of discovering a web site's internal pages include, for instance, exhaustively crawling the web site, collecting links posted on blogs and/or social media, and gathering frequently visited internal pages from site-traffic metrics maintained by web sites or reported by browsers. We preferred search engine results as they combine all three of the above approaches: Search engines routinely crawl web sites exhaustively (except pages disallowed via `robots.txt` [60]), collect links posted on other web sites to rank results (e.g., Google's PageRank [91]), and track internal pages frequently searched and visited by users [70]. Also, more than two-thirds of "trackable" web

traffic comes from search engines [19], where trackable implies that the user reached the concerned web page from another web site (as opposed to entering the page's URL directly in the browser or clicking a bookmark). Additionally, search engine results have empirically proven to be fairly stable, and stability is a desirable property of a top list [67].

On the stability of $\mathcal{H}_{2K}$

$\mathcal{H}_{2K}$ is different from existing top lists in that it has a two-level structure: web sites at the top and the web pages (or URL sets) of those sites at the bottom. The top level will inherit, naturally, the stability (or *churn*) of the top list used for bootstrapping—$\mathcal{A}_{1M}$ in this case. We observe, for instance, a 20% mean weekly change in the web sites that appear in $\mathcal{H}_{2K}$. This change is directly inherited from the Alexa top 5K list, a subset of $\mathcal{A}_{1M}$, that was used to bootstrap $\mathcal{H}_{2K}$. Prior work also observes that the Alexa Top 5K list experiences about 10% *daily* change [93].

Additionally, $\mathcal{H}_{2K}$ may also experience a churn at the bottom level: The set of N (internal-page) URLs, selected from search results, in each URL set at the bottom level may change over time. We estimate the weekly churn as the fraction of (internal-page) URLs present in the list on week i, but not on week $i+1$ for web sites present on both weeks. In computing this churn, we assume no ordering among the pages at the bottom level; although the search results are ranked, we advise against assigning any meaning to the ordering of the URLs in a URL set.[1] We use this weekly churn to characterize the stability of $\mathcal{H}_{2K}$.

Across a 10-week period starting in February 2020, $\mathcal{H}_{2K}$ experiences a 30% weekly churn in the internal pages at the bottom level. It is not surprising that inclusion of internal pages introduces additional churn: `nytimes.com` consistently remains a popular web site, but its news headlines change multiple times in a day. Perhaps the churn in internal pages is even desired as the list should ideally reflect the changing internal states of the web sites it is representing. By comparison, a subset of $\mathcal{A}_{1M}$ of the same size as $\mathcal{H}_{2K}$, Alexa top 100K, experiences a mean weekly change of 41% in web sites over the same period. The higher churn in $\mathcal{A}_{1M}$ has not been a hindrance to using it in web-perf. studies. If the churn in internal pages in $\mathcal{H}_{2K}$ is deemed too high, we can improve the list's stability by using the same techniques that are used to improve the stability of top lists—averaging the results over longer periods of time as Pochat et al. suggest [84].

3.1 The $\mathcal{H}_{1K}$, $\mathcal{H}_{t30}$, $\mathcal{H}_{t100}$, and $\mathcal{H}_{b100}$ Lists

For the purposes of this study, we created a smaller version of *Hispar*, namely $\mathcal{H}_{1K}$, with 1000 web sites. We bootstrapped $\mathcal{H}_{1K}$ using the $\mathcal{A}_{1M}$ list downloaded on March 12, 2020. The URL set for each web site in $\mathcal{H}_{1K}$ consists of one landing page and at most 19 internal pages, comprising 20,000 web pages in aggregate; if the search for internal pages on a site revealed less than 5 results, we dropped that site. In addition to making the measurements more tractable, the smaller top list makes our experiments similar to that of studies that received a "major" revision score in our survey: 60% of such studies use 1000 or fewer web sites, and 77% use 20,000 or fewer web pages. To demonstrate how the differences between the landing and internal pages change based on the popularity ranking of web sites, we also use three different subsets of $\mathcal{H}_{1K}$: The lists $\mathcal{H}_{t30}$ and $\mathcal{H}_{t100}$ consist of the URL sets of the top 30 and 100 web sites, respectively, of $\mathcal{H}_{1K}$, while the $\mathcal{H}_{b100}$ contains those of the bottom 100 web sites of $\mathcal{H}_{1K}$.

We fetched web pages in $\mathcal{H}_{1K}$ by automating the Mozilla Firefox browser (version 74.0), using the tools from Enghardt et al. [38]. We performed the page fetches from an Ubuntu 18.04 server with an Intel 8-core i7 processor and 32 GB of RAM. We shuffled the set of all landing pages, iterated over this set 10 times, and fetched each page with an *empty* cache and *new* user profile. We then shuffled and fetched the internal pages in a similar manner, except we fetched the internal pages for each web site only once.[2]

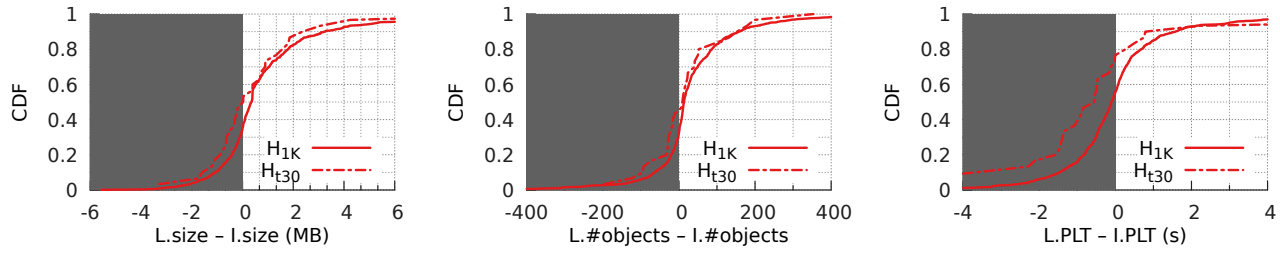
After each web-page visit using the automated browser, we collected the HTTP Archive (HAR) files [104] from the browser and data from the Navigation Timing (NT) API [105]. The HAR files provide various details (e.g., response size and time) about resources fetched when loading a page, while the NT data provides performance measures of the web page fetch and load. To compare and contrast different characteristic features of internal pages with those of landing pages, we typically compute the cumulative distribution function (CDF) of each such feature for each of the two page types and compare these CDFs. The "landing" CDF is computed over 10,000 values, while the "internal" CDF is computed over 19,000 values. For each such comparison, we also present the p-values ($\mathcal{D}$) from a two-sample Kolmogorov–Smirnov test [85], with the null hypothesis of the test being that the CDFs are not significantly different (i.e., they have both been drawn from the same underlying distribution). A low $\mathcal{D}$ value, hence, indicates a high statistical significance, i.e., it is less likely that the samples were drawn from the same distribution.

Ethical considerations. Our measurements do not involve real users or include any personally identifiable information. When gathering these measurements, we avoided exhaustive crawls of web sites to induce minimal load on the web servers and infrastructure. Measurements over $\mathcal{H}_{1K}$ involve 30 page fetches per web site, spread over 5 days. For the limited-exhaustive-crawl experiments in §4, we fetched 500 pages each for 5 web sites. These fetches were also spread out such that there was at least a 5-second gap between consecutive page fetches. In the unlikely scenario where our web-page fetches impose undue load on a web server, we took measures to facilitate web-site owners or administrators to opt out of our experiments. To this end, we modified the HTTP `User-Agent` header of our automated browser to include a URL pointing to our project web page. The project web page describes who we are, the intent behind the crawl, and a procedure to *opt out* of the crawls. We did not, however, receive any opt-out requests, presumably because our crawl volumes were negligible for an Alexa-ranked web site.

Limitations. First, our methodology has a sample bias of selecting the top roughly 2000 web sites from the $\mathcal{A}_{1M}$ list. We also note that the magnitude of the differences we observe may not generalize well to less popular web sites. Second, we do not measure internal pages that are behind a user log-in, such as the Facebook news feed.

[1] Search engines do not reveal the exact metric by which the search results are rank ordered.

[2] Our intent is to compare the observations in two categories—landing and internal pages. The number of individual samples in the latter suffices to capture the variance in observations, making it unnecessary to repeat the fetches.



(a) (b) (c)

Figure 2: ***Overview of differences between landing (L) and internal (I) pages: For web sites in $\mathcal{H}_{1K}$ ($\mathcal{H}_{t30}$), (a) 65% (54%) have landing pages that are larger than the median size of their internal pages; (b) landing pages of 68% (57%) have more objects; and (c) page-load times of landing pages are smaller for 56% (77%).***

Such pages may drastically differ from the landing page as well as other internal pages. Third, we measure all pages with a "cold" browser cache, which means that objects fetched while loading the landing page of a web site do not affect the loading times of internal pages that may also host a subset of these objects. Lastly, whether most users navigate to internal pages through the landing page, or through direct links on search engine results and other web sites, remains unknown to us.

4 OVERVIEW OF DIFFERENCES

While two arbitrary web pages from the same web site may share some common objects and resources, they may have also significant differences in structure, size, and content. In this section, we present the high-level differences between landing and internal pages that we discovered.

Differences in size and object count

We begin with a focus on two questions: *(a) Are landing and internal pages similar in size, on average?* and *(b) Do landing and internal pages have similar structure, on average?*

We defined the size of a web page as the aggregate size of the constituent objects (i.e., the sum of sizes of all entries in the corresponding HAR file) comprising that page. For each web site ω in $\mathcal{H}_{1K}$, we measured the difference in size between the landing page (i.e., the median size observed across 10 page loads) and the median size of the internal pages. The CDF of these differences, in Fig. 2a, indicates that for 35% of the sites, the landing pages are smaller in size (the shaded region) than the internal pages. For 5% of the web sites, the median sizes of internal pages are at least 2 MB larger than the landing pages, while for another 20% they are at least 2 MB smaller than the landing pages. The geometric mean of the ratios of page sizes of landing to internal pages reveals that landing pages are, on average, 34% larger than internal pages. The size differences also vary substantially with the rank of web sites (see Fig. 9b in Appendix A).

We used the number of objects in a web page (i.e., the number of entries in the corresponding HAR file) as a crude approximation of its structure. Then, to estimate the structural differences between the two page types, we computed the difference between the number of objects on the landing page and the median number of objects across internal pages for each web site in $\mathcal{H}_{1K}$. Fig. 2b shows the CDF of these differences, indicating that the two page types are significantly different from one another. For 32% of the sites, landing pages have fewer objects (the shaded region) than internal pages. Comparing the shaded region of this plot with that of Fig. 2a reveals that for 5% of the web sites, the landing pages, despite having fewer objects, are larger than their corresponding internal pages. The geometric mean of the ratios of object counts of landing to internal pages indicates that landing pages have 24% more objects, on average, than internal pages. The differences in object count, as with size, vary with web site rank: In $\mathcal{H}_{t30}$, 57% of web sites have landing pages with higher object count than internal pages, but that number jumps to 68% in $\mathcal{H}_{b100}$ (see Fig. 9c in Appendix A).

Differences in page load and render times

Now, we turn our attention to two widely used performance metrics and ask, "*Is the time to load and render a page, on average, similar between landing and internal pages?*"

We define the page-load time (PLT) of a web page as the time elapsed between when the browser begins navigation (i.e., when the `navigationStart` event fires) and when it renders the first pixel (i.e., when the `firstPaint` event fires). Then, we estimated the performance difference between the two page types, for each web site in $\mathcal{H}_{1K}$, using the difference between the PLT of the landing page and the median PLT of the internal pages. Landing pages in $\mathcal{H}_{1K}$ are heavier (Fig. 2a) and have more objects (Fig. 2b). Since these parameters have implications for PLT [25], we expect landing pages to have higher PLTs than internal pages. We observe, nevertheless, the opposite: Landing pages load faster than internal pages for 56% of the web sites (Fig. 2c). These differences also vary significantly based on the web site rank: 77% of web sites in $\mathcal{H}_{t30}$ have faster landing pages than internal pages, but that percentage drops to 59% in $\mathcal{H}_{b100}$ (Fig. 9a in §A). One reason that landing pages load faster than internal pages could be that resources in landing pages are more likely to be cached at a CDN (see §5.1), since they are also likely to be relatively more popular (i.e., more frequently requested by users). It could also be that web developers optimize the landing-page design more meticulously than the internal pages, to avoid frustrating or distracting end users with a slowly loading landing page. We explore these questions in §5.

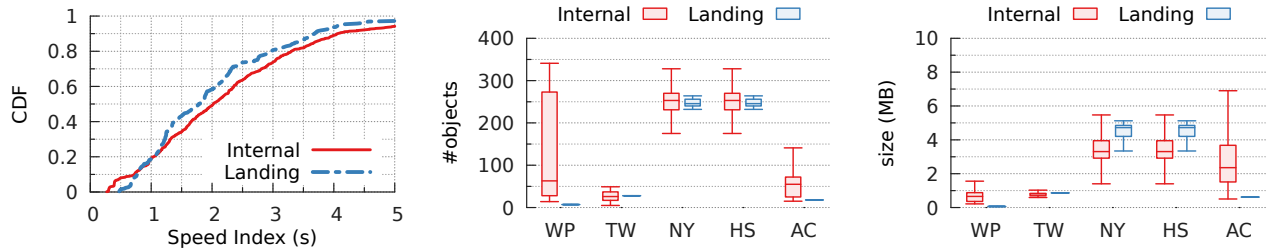


Figure 3: ***(a) Content on landing pages displays, in the median, 14% faster than that on internal pages ($\mathcal{D}$=0.01). Limited exhaustive crawls of five web sites, Wikipedia (WP), Twitter (TW), New York Times (NY), HowStuffWorks (HS) and an academic web site (AC), show that internal pages differ from landing pages and from each other in (b) number objects and (c) page size.***

To address the well-known shortcomings in PLT [34, 38], we also measured the SpeedIndex (SI) scores [46, 110] of the two page types using Google's PageSpeed Insights API [44]. The SI score measures how quickly the content on a web page is visually populated [110]. A low SI score indicates that the page loads quickly. For each web site in $\mathcal{H}_{t30}$, we computed the median SI scores of the page types as follows. We derived the SI scores ten times for the landing page and computed the median. For internal pages, we derived the score once for each of the 19 pages, and computed the median of all pages. Fig. 3a shows that content on internal pages visually loads 14% more slowly than that on landing pages in the median.

Limited exhaustive crawl

We supplemented the above experiments with an exhaustive crawl of five web sites. We selected wikipedia.org (WP), twitter.com (TW), nytimes.com (NY), howstuffworks.com (HS), and csail.mit.edu (AC), with Alexa ranks 13, 36, 67, 2014, and "unranked" respectively. We crawled the landing page of each web site and followed links to internal pages recursively until we obtained at least 5000 unique URLs for each domain. We fetched the landing pages ten times and computed the medians of relevant metrics. We randomly sampled 500 URLs from the discovered internal pages, and fetched them once. The internal pages of these web sites show a large variation in object counts (Fig. 3b) and page sizes (Fig. 3c). Per these figures, internal pages differ substantially not only from landing pages, but also from one another. The distribution of object counts and page sizes shows that our inferences would not change significantly for a random subset of 19 internal pages; as such a random selection would likely not change the median values. Analyzing only 19 internal pages and using the median values of object counts, page sizes, and PLTs, only limits the magnitude of these differences.

In summary, the differences between landing and internal pages of a web site are *not* random. Averaging the results of an analysis over a large number of landing pages is unlikely to eliminate the inherent bias in the differences; besides, these differences vary across popularity ranks. These differences narrow the scope of studies that rely only on landing pages; virtually all such studies would have to revise their claims and insights to generalize to all web pages and not just landing pages. We delve deeper into the differences between the landing and internal pages and highlight how they affect prior web-perf. studies in §5 and §6.

5 CONTENT AND DELIVERY

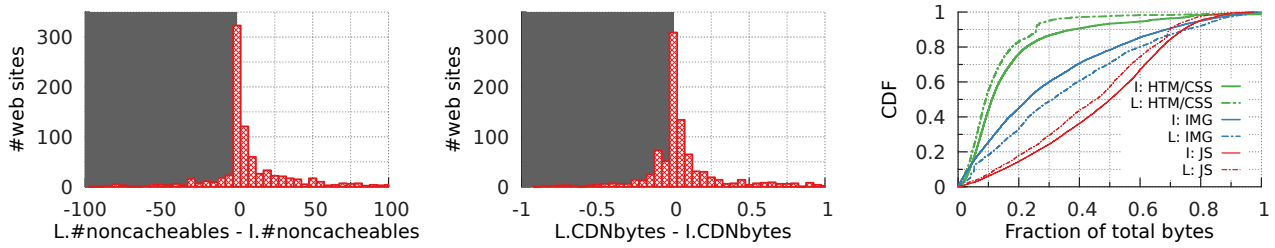
In this section, we analyze the differences between landing and internal pages in content and in optimizations used for delivering content quickly. When discussing these differences we also highlight the implications for select prior web-perf. studies.

5.1 Cacheability

Caching is the most commonly used technique to improve web page performance. Rather than serving content to users from origin (i.e., publisher's) servers, caching attempts to serve them from servers in close proximity to the users. The round-trip time (RTTs) of the network path between the end points in case of the latter is typically shorter than that of the former. The majority of web content today is delivered via content delivery networks (CDNs) [31], which often act as intermediate caches along the path between users and origin servers. The number of cacheable objects constituting a web page and the volume of data delivered through a CDN, hence, have a large impact on web page performance.

To count the cacheable objects in landing as well as internal pages we analyzed the HAR files generated after fetching a given web page. We used the HTTP `request method` and `response code` to check whether an object is cacheable [71]. While we cleared the web-browser cache prior to every page fetch, the state of intermediate caches (i.e., along the cache hierarchy between the browser and the server of the content provider) could change depending on the cacheability of objects. Where the objects of a web page are served from—the content provider or an intermediate cache—does not, however, affect the count of (non-)cacheable objects.

Fig. 4a plots the CDF of the number of non-cacheable objects: Landing pages in $\mathcal{H}_{1K}$ have, in the median, 40% more non-cacheable objects than internal pages. If we measure cacheability, however, as the fraction of cacheable bytes to the total bytes in the page, both landing and internal pages have similar cacheability. Landing pages, hence, have more (non-)cacheable objects by virtue of simply having more objects in general. The lower PLTs of landing pages compared to internal pages (refer Fig. 2c), therefore, are not due to the former having more cacheable content than the latter. This differential between landing and internal page cacheability also implies that the effectiveness of a performance optimization such as caching cannot be generalized to all pages; internal pages might not benefit as much from caching (or CDNs) as landing pages.



(a) (b) (c)

Figure 4: ***For web sites in $\mathcal{H}_{1K}$: (a) 66% have landing (L) pages with more non-cacheable objects than their internal (I) pages (40% more in the median); and (b) landing pages of 57% have a higher fraction of bytes delivered via CDNs (13% more in the median). Internal pages have, in the median, 10% more JS bytes as a fraction of total bytes, 36% less image bytes, and 22% more HTML/CSS bytes than landing pages ($\mathcal{D} \ll 0.00001$ for HTML/CSS, image, and JS bytes)***

To determine whether a particular HTTP request was served through a CDN, we used multiple heuristics (e.g., domain-name patterns, HTTP headers, DNS `CNAME`s, and reverse DNS lookup). We obtained these heuristics from [99]. Although these heuristics are neither exhaustive nor necessarily accurate, simply ascertaining whether the object was delivered by a well-known CDN suffices for this study. In the web-page-fetch experiments with $\mathcal{H}_{1K}$, we identified more than 40 different CDNs using these heuristics. Then, we measured the ratio of bytes delivered by CDNs to the total size of the page.

We observe that in the median, the fraction of bytes delivered via CDNs for internal pages is 13% lower than for landing pages. Whether a given object (request) experienced a cache "hit" or "miss" at the CDN does not have any implications for the above observation, but it affects the performance (or load times) of the pages. We suspect that landing pages have more "popular" (i.e., frequently requested by users) objects, which might lead to a higher cache hit ratio at the CDN. Measuring whether an object request experienced a cache hit presents, unfortunately, several challenges: Our measurements are not spread over a long time period; We only measure from a single vantage point, so our visibility is limited to the CDN site closest to the vantage point at the time of measurement; and, lastly, the mechanism through which CDNs report a cache hit or miss is not standardized. We used, nevertheless, the HTTP `X-Cache` header (used by at least two major CDNs [3, 39]), to identify whether a request experienced a cache hit or miss. For web sites in $\mathcal{H}_{1K}$, we found that cache hits for landing-page objects are 16% higher than those for internal-page objects, suggesting that internal pages do not benefit as much from CDNs as landing pages.

Implications for prior work. Vesuna et al. showed that PLTs of web pages on mobile devices do not benefit as much from improvements to cache hit ratios as desktop devices [103]. Using 400 landing pages selected uniformly at random from the Alexa Top 2K list, they show that a perfect cache hit ratio, compared to no caching, would reduce PLT by 34% for desktop devices, but only 13% for mobile devices. We find that landing pages of sites in $\mathcal{H}_{1K}$ have more non-cacheable objects than internal pages; the differences are also not uniform over the popularity ranks of web sites in $\mathcal{H}_{1K}$ (refer §A). Depending upon the particular random subset of 400 landing pages selected, Vesuna et al. could underestimate to various degrees the effect of caching on PLT for those web sites. Similarly, Narayanan et al. evaluate a new cache-placement algorithm for CDNs using only the landing pages of 83 randomly chosen web sites from the Alexa Top 1K (40%) and Alexa Top 1M (60%) lists [86]. They report that their algorithm can reduce PLT by 100 ms or more for 30% of the web pages. Per Fig. 4b, internal pages have 13% less content delivered via CDNs than landing pages, and such content will, naturally, draw no benefits from Narayanan et al.'s placement algorithm. They perhaps overestimate the PLT decrease, since they did not consider internal pages in their evaluation.

5.2 Content Mix

The discussion on cacheable objects leads to a broader and a more general question: Do the landing and internal pages differ substantially in terms of the distribution of the different types of objects they comprise?

To estimate the distribution of different types of objects constituting the web pages in $\mathcal{H}_{1K}$, we gathered the MIME types [106] of objects from the HAR files. We collapsed them into *nine* categories (`audio`, `data`, `font`, `HTML/CSS`, `image`, `JavaScript`, `JSON`, `video`, and `unkown`) to simplify the analyses. We then measured, for each web site, the relative size of content (i.e., as a fraction of total page size) in each category. Fig. 4c shows the distribution of the relative size of content in three different categories. The other six categories combined only contribute 6% (7%) of the bytes for landing (internal) pages, and, hence, we omitted them for clarity.

Landing and internal pages of web sites in $\mathcal{H}_{1K}$, per Fig. 4c, differ substantially in terms of HTML and Cascading Style Sheets (HTM/CSS), JavaScript (JS), and image (IMG) contents. Internal pages, in the median, have 50% JS content ("I: JS") while landing pages have 45%—a 10% change. Landing pages also have 22% less HTML/CSS content than internal pages as a fraction of total bytes. Conversely, landing pages' fraction of image bytes ("I: IMG") is 36% higher than that of internal pages. These differences could partly be due to landing pages typically having a few large images (e.g., banners) or many small images (e.g., thumbnails of photos related to various stories on a news web site). The smaller JS content of landing pages could be due to web-designers intending to keep them simpler (i.e., fewer computations to be processed by the browser), thereby helping them load faster than internal pages. Internal pages,

in contrast, by virtue of containing more JavaScript might load slower than landing pages, even if they each have the same number of objects and page size as the corresponding landing page. The significant differences in contents between landing and internal pages highlight, once again, that techniques for optimizing landing pages (which were the focus of virtually all prior work on web performance optimization) might not be effective, or even feasible, for internal pages.

Implications for prior work. Butkiewicz et al. conducted a large-scale measurement study to analyze the complexity of web pages and investigate the implications for performance measurement [25]. They tested the landing pages of 2000 web sites randomly selected from the Alexa Top 20K list, and reported that, in the median, JavaScript constituted 25% of the page size. We find (in Fig. 4c) that JavaScript contributes, in the median, to 45% of a landing page's size,[3] but that contribution increases to 50% in internal pages. Therefore, ignoring internal pages in measurement studies such as [25] would underestimate the amount of JavaScript on the web and overestimate the amount of multimedia (e.g., images, audio, and video) content. Performance optimization efforts that rely on such measurement studies could in turn propose misleading guidelines for performance improvements.

5.3 Multi-Origin Content

We refer to the content on a web page served from two or more domains as multi-origin content. The landing page of www.nytimes.com uses, for instance, the domain `static01.nyt.com` for serving images, `cdnjs.cloudflare.com` for objects delivered by the Cloudflare CDN, `use.typekit.net` and `fonts.gstatic.com` for fonts, `ad.doubleclick.net` for placing ads via Google DoubleClick, and `www.google-analytics.com` for serving scripts for analytics, to name a few. The request to load such a page will, if the browser's DNS cache is empty, result in the browser issuing many DNS queries, one for each unique domain. Discrepancies in the prevalence of multi-origin content between landing and internal pages may have implications for their performance (e.g., PLTs).

We estimated the prevalence of multi-origin content in web pages in $\mathcal{H}_{1K}$ by counting, for each page fetch, the number of unique domains encountered across all requests issued by the browser to load that page's contents. Multi-origin content is more prevalent in landing pages in $\mathcal{H}_{1K}$ than in internal pages (Fig. 5): The former has 29% more unique domain names, in the median, than the latter. The magnitude of these differences also varies over web site popularity ranks (see §A).

The difference in the number of DNS queries issued between landing and internal pages, owing to the discrepancies in the prevalence of multi-origin content, might, however, be masked by the local resolver if the DNS responses for the domains queried are usually found in the local resolver's cache. Therefore we estimated the "hit" rate of a resolver's cache as follows. We picked the top 5K most popular (i.e., most frequently observed) domains from the Cisco Umbrella list [54], and issued two consecutive queries to our local (ISP's) resolver as well as to Google's public DNS resolver.

[3]The increase is roughly in line with the increase in JavaScript bytes that HTTP Archive reports since 2011 [53].

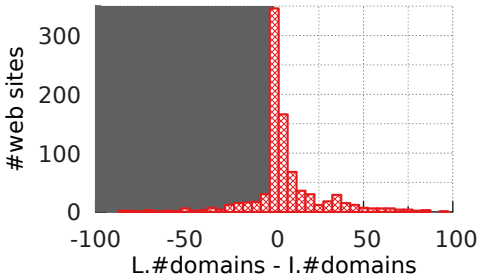


Figure 5: *For web sites in $\mathcal{H}_{1K}$, 67% have landing pages that fetch content from more origins (29% more in the median).*

For each resolver, if the response time for the second query was significantly lower than the first, we label the first as a cache "miss" at that resolver; otherwise, we mark the first response as a cache "hit." For the most popular 5K domains on the Internet, we observed a hit rate of only about 30% at our local resolver and about 20% at Google's public resolver. The low cache hit rates we observe are in line with previous studies [1, 4], and are mostly explained by the practice of setting low time-to-live values for request routing [73] and cache fragmentation at the Google resolver [49]. Caching at DNS resolvers, hence, does not completely mask the performance impact of multi-origin content.

Implications for prior work. Böttger et al. explored the performance cost of using *DNS over HTTPS* (DoH) [52] for web browsing [17]. They crawled the landing pages of Alexa Top 100K web sites, recorded the number of DNS requests required to fetch each page, measured the overhead incurred by DoH with respect to the traditional DNS over UDP, and measured the difference in PLT resulting from those overheads. They observe, in the median, 20 DNS requests per landing page for web sites in the Alexa Top 100K. We observe, however, that for most sites in $\mathcal{H}_{1K}$ landing pages fetch content from more origins than internal pages. This difference also varies over web site popularity ranks. If Böttger et al.'s study was generalized to internal pages, it would overestimate the count of DNS requests per page, and consequently miscalculate the cost of switching over to DoH.

5.4 Inter-Object Dependencies

Downloading and rendering the objects on a web page often requires the web browser to handle complex dependencies between the objects. A browser might have to fetch, for instance, two objects—say, a JavaScript and a CSS file—and parse them before fetching a third object (say, an image). Such relationships between objects are encoded typically in a data structure called the *dependency graph* [66, 76, 108]. Nodes in the graph represent the objects and the directed edges encode the dependencies between them. We define the *depth* of an object as the shortest path from the root document to that object. Every internal node on this path is another object, which must be downloaded before the download of the concerned object begins, slowing down the page load process. On any given page, there is only one object at depth 0—the root HTML. All object fetches that the root HTML triggers lie at depth

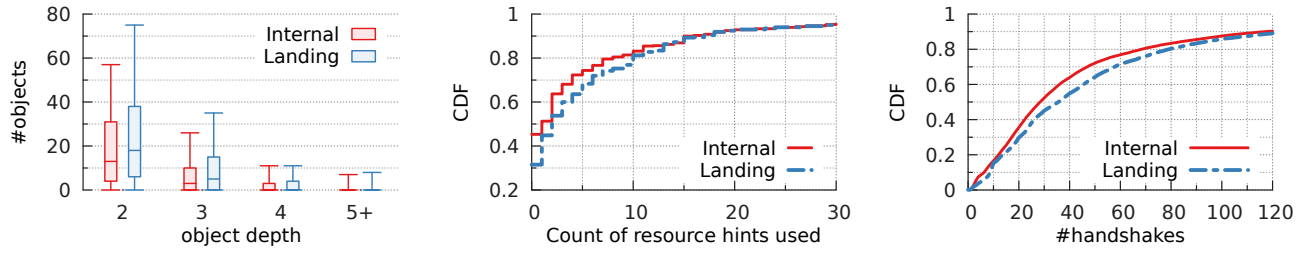


Figure 6: ***(a) Landing pages have more objects at each level of depth than internal pages do. In the median, landing pages have 38% more objects at depth 2. (b) 69% of landing pages use at least one HTML5 resource hint, whereas 45% of internal pages have no hints ($\mathcal{D} \ll 0.00001$). In the median, (c) landing pages perform 25% more handshakes than internal pages do ($\mathcal{D} \ll 0.00001$)***

.

1. Below, we analyze objects at depths 2 and greater. (There are far more objects at depth 1 than at any other depth.)

To analyze the differential between internal- and landing-page object depths, we fetched the web pages of web sites in $\mathcal{H}_{t100}$ and $\mathcal{H}_{b100}$, and generated the dependency graph of each page using the tool from [28]. This tool uses the Chrome DevTools Protocol [29] to track which object triggers which other object fetches (via the `initiator` parameter in the `requestWillBeSent` event) and build the graph. We measured the depths of all objects, and compared the number of objects on landing and internal pages at each depth. Fig. 6a shows that landing pages have consistently (i.e., in the 50th, 75th, and 90th percentiles) more objects than internal pages at depths 2 and 3; at depths 4 and beyond, even if the medians are 0, the former has more objects than the latter in the tail (90th/95th percentile). Landing pages, by this metric, have a more complex page structure than internal pages.

Implications for prior work. There exists a rich body of prior work that employs one or more variants of the dependency graph to determine how to improve page-load times [26, 76, 92, 109]. These optimizations generally work by delivering objects at greater depths of the dependency graph earlier than the browser would normally fetch them. Complex dependency graphs present the most opportunities for these optimization efforts [76]. For evaluation, these works use only the landing pages from some subset of the Alexa Top 1M, and ignore internal pages. We used a rudimentary approach to building dependency graphs as the tools used in these studies were not available. Our measurements nevertheless suggest that landing pages have more complex dependency graphs than internal pages. By ignoring internal pages, the aforementioned efforts, hence, may have overestimated the impact of their optimizations.

5.5 Resource Hints

The interdependencies between objects (§5.4) and the fact that web pages often serve objects from many different domains make it hard for a web browser to determine how to optimize the fetch process and load pages quickly. HTML5 *resource hints* are a recent attempt to remedy this problem. Resource hints are primitives that provide hints to the web browser such as which domains it should connect to (via the `dns-prefetch` and `preconnect` primitives), which resources it should fetch (via `prefetch`), and which ones to preprocess (via `prerender`) in parallel [107]. These web-developer-provided hints can, if used correctly, result in significant web page performance improvements.

We inspected the HTML DOM of the web pages in $\mathcal{H}_{t100}$ and $\mathcal{H}_{b100}$ and counted the number of resource hints used in each page. Fig. 6b shows the CDF of the counts of resource hints used in internal as well as landing pages. For the web sites measured, we find that the use of resource hints is more prevalent in landing pages than internal pages: 69% of landing pages use at least one resource hint, whereas 45% of internal pages have no hints. This discrepancy in resource-hint use is even larger for web sites in $\mathcal{H}_{t100}$: 52% of internal pages in $\mathcal{H}_{t100}$ don't use any hints.

Implications for prior work. We did not find any large-scale study of the implications of resource hints for page performance, but there exists anecdotal evidence on how the use of HTML5 resource hints reduces PLT [80]. Evidently, if a study is conducted in the future, it would overestimate the prevalence and performance of these hints if it only considered landing pages. The discrepancy in resource-hint use also suggests that web developers are optimizing landing pages more carefully. Since internal pages of more than 90% of web sites in $\mathcal{H}_{1K}$ host or serve content on more than one domain, web developers *must* at the very least consider using the `dns-prefetch` hint for internal pages. Future work can use our publicly available lists to carefully evaluate which hints could help internal pages, and to what extent.

5.6 Round-trip and Turnaround times

The number of objects on a web page correlates with its page-load time (PLT) [25], because the browser must fetch over the network each object that is not already in its cache. Ideally, the browser would fetch all of these objects in parallel, and the PLT would ultimately depend only on the time it takes to download the largest object, or simply on bandwidth. Web pages have, however, objects with complex inter-dependencies (see §5.4) and latency directly affects their PLT [30]: The browser downloads and parses an object, discovers other objects that depend on it, and only then starts to fetch them. HTTP/2 Server Push [13] and QUIC [63] are among the recent efforts towards increasing parallelism in fetching objects and reducing the number of round-trips required to fetch objects, and thus decreasing PLTs. Since the landing and internal pages of

a web site significantly differ in structure and contents, the efficacy of these optimizations will also vary between the two page types.

We used the HAR files from our page-fetch experiments to analyze the time spent in downloading the different objects on each page in $\mathcal{H}_{1K}$. The HAR file breaks down the time spent in downloading each object into seven steps: (1) `blocked`, (2) `dns`, (3) `connect`, (4), `ssl`, (5) `send`, (6) `wait`, and (7) `receive`.[4] We treat the combined times of `connect` and `ssl` as the total time spent in TCP and TLS handshakes, and `wait` as server processing time. Since the browser typically fetches many objects in parallel, the sum of handshake times for all objects is only an approximation of the effect of handshakes on PLT.

We observe that on average, the browser spends the same amount of time in handshakes prior to downloading an object regardless of page type. Also, the fraction of objects fetched over new connections, which require a handshake, is nearly the same on landing and internal pages. However, landing pages typically have more objects (cf. §4) and multi-origin content (see §5.3) than landing pages. As a consequence, landing pages in $\mathcal{H}_{1K}$ spend 28% more time, in the median, performing handshakes than do internal pages. They also perform, in the median, 25% more handshakes than internal pages do (Fig. 6c). Per these observations, internal pages would benefit less than landing pages from efforts that reduce the number of round-trips involved in a handshake, such as QUIC [63], TCP Fast Open [89], and TLS 1.3 [90]. Ignoring internal pages in the evaluation of such optimizations could exaggerate their benefits.

In our experiments, about half of the time it takes to download an object is, on average, spent in the `wait` step. A browser's request to fetch an object might have to spend time in `wait` for several reasons, e.g., *stalls* or processing delays at the server, and queuing delays along the route. We find that objects in internal pages spend 20% more time, in the median, than those in landing pages (Fig. 7). This finding combined with the earlier observation that internal-page fetches result in more cache "misses" at the CDN (cf. §5.1) suggests that the larger `wait` times are perhaps due to the *turnaround* or processing times at the CDN servers. CDNs have a complex hierarchy of servers acting as a multi-level cache to quickly and efficiently serve objects to users. Since most objects on a web page are rather small [63] and the connections between different CDN servers as well as those between the CDN and origin (or content-providers') servers are typically persistent [96], the time to download an object is dominated by the round-trip time between the CDN servers or between the CDN server and the origin server. These findings suggest that internal pages induce more back-office web traffic than landing pages at the CDN, and are, thus, affected more by the latency experienced in the CDN backhaul.

Implications for prior work. Research efforts over the past few years have renewed the networking community's interests in understanding and improving latency in the Internet backbone [15, 18, 95]. There have also been other efforts that focus on minimizing the number of round-trips in the upper layers of the network-protocol stack [13, 63, 89, 90]. None of these efforts, however, point out how landing pages are already significantly faster than internal pages despite the former being heavier than the latter. They do not examine how physical-layer latency improvements or protocol optimizations would help in speeding up the slowest parts of the web, comprising the internal pages of web sites. Recent follow ups on measuring the performance impact of QUIC and other protocols [111, 112] also ignore internal pages. Having both the design and the evaluation parts of web performance optimization efforts completely ignore internal pages can be dangerously misleading to research as well as practice.

[4] We refer the reader to [104] for a detailed description of these steps.

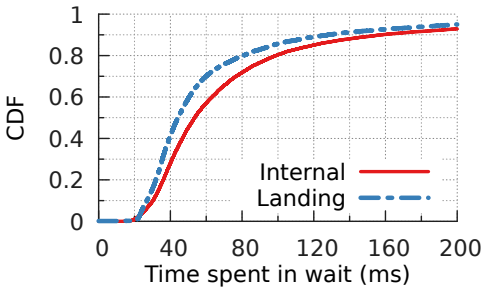


Figure 7: *Objects on internal pages spend 20% more time in* `wait` *than those on landing pages ($\mathcal{D} \ll 0.00001$).*

6 SECURITY AND PRIVACY

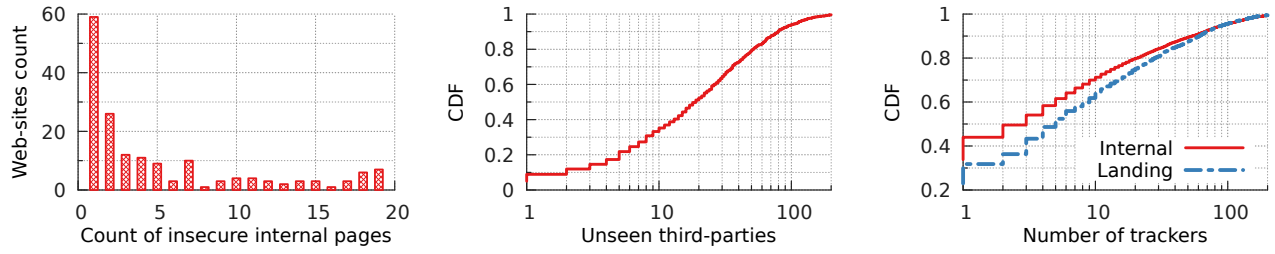
Below, we discuss how including the internal pages of web sites could affect analyses pertaining to security and privacy of the web. As in prior sections, we follow up our observations with implications for relevant prior work.

6.1 HTTP and Mixed Content

The use of (cleartext) HTTP for serving web sites has well-known security pitfalls, e.g., session hijacking and man-in-the-middle attacks. Owing to a concerted effort from developers, content providers, and web browsers, the majority of web content today is served over (secure) HTTPS [48]. There are numerous ongoing efforts to further improve the users' security and privacy through technologies like Certificate Transparency (CT) [64] and HTTP Strict Transport Security (HSTS) [51]. In this section, we simply ask whether the security attributes of landing pages are similar to those of internal pages.

We found only 36 of the 1000 web sites in $\mathcal{H}_{1K}$ to serve their landing pages over HTTP; the rest redirected to HTTPS versions. Among the web sites with secure landing pages, we discovered that 170 web sites had at least one HTTP internal page (Fig. 8a). In most cases, the same domain was hosting the non-secure internal page (e.g., http://www.fedex.com/us/international/) while in others, a seemingly secure internal page was found redirecting to a non-secure page on a different domain (e.g., https://www.amazon.com/birminghamjobs redirected to a plaintext page on `amazon.jobs`, which has now moved to HTTPS.) We regret such poor practices, particularly in well-known web sites such as amazon.com, ebay.com, and adobe.com.

We also analyzed the web sites in $\mathcal{H}_{1K}$ for pages hosting *mixed* content. A web page served over HTTPS is called a mixed-content page if it includes content fetched using (cleartext) HTTP. The presence of mixed content could undermine the security of a web page. Mixed content may, for instance, leak information about a



(a) (b) (c)

Figure 8: ***For web sites in $\mathcal{H}_{1K}$, (a) 170 have secure landing pages but at least one non-secure internal page, 36 have 10 or more; (b) internal pages collectively contact, in the median, 18 third parties that are never seen on the landing page of the same web site; (c) at the 80th percentile, landing pages make 40% more tracking requests ($\mathcal{D} \ll 0.00001$).***

user's browsing behavior, and expose content to man-in-the-middle attackers. Some web browsers flag such pages by showing a visible warning to the user and some simply refuse to load the page. We searched for only "passive" mixed content (i.e., when images and other static resources are served using HTTP on an HTTPS page), since "active" mixed content (e.g., JavaScript) is blocked by default on most browsers [27]. We found that while only 35 web sites in $\mathcal{H}_{1K}$ have landing pages with (passive) mixed content, 194 have at least one mixed-content internal page. Since we fetched only 19 internal pages per web site in $\mathcal{H}_{1K}$, our estimates of the prevalence of (cleartext) HTTP and mixed content are probably only the lower bounds.

Implications for prior work. There have been numerous studies on HTTPS adoption using various data sources such as top lists [61], DNS zone files [10], Certificate Transparency logs [102], port scans [35], and real user traces [40]. Unfortunately, all these data sources except for the real user traces exclude internal pages of a web site. Felt et al. studied HTTPS adoption, using real traces as well as other data sources, among popular web sites using top lists [40]. Paracha et al. recently studied the prevalence of non-secure internal pages in web sites with secure landing pages, and content differences between HTTP and HTTPS versions [82]. We did not find any prior work on the prevalence of HTTPS pages redirecting to HTTP pages.

6.2 Third-Party Dependencies

Modern web pages depend on a large number of third-party content including, but not limited to, static content served from a CDN, analytics, and advertising. Such dependencies offer a lucrative attack vector for malicious actors to, for example, take down a large portion of the web by compromising one entity. A web page on domain a could access third-party content on domain b, which in turn could depend on other third-party content on domain c. These third-party dependencies can be encoded in the *dependency chain* $a \rightarrow b \rightarrow c$. Below, we investigate if the landing page sufficiently represents the dependency chains for a web site.

The domain in the URL of an object on a page is considered a third-party domain if it does not belong to the same second-level domain (SLD) as the page being fetched. For example, for a web page on `www.guardian.com`, `cdn.akamai.com` is a third-party domain, but `images.guardian.com` is not. We take public (domain) suffixes into consideration to ensure that, for instance, `tesco.co.uk` will be a third-party domain for `bbc.co.uk`. Our method is prone to false positives in case of the same organization owning different domains: `microsoft.com` is counted as a third-party on `skype.com`. Such false positives should, however, be similar for both page types, and thus would not introduce a systemic bias.

For this study, we focused only on the unique third-parties involved and ignore the dependency relationships among them. We counted, hence, the number of unique third-party domains observed on at least one internal page but never on the landing page. We find that, in the median, internal pages collectively fetch content from 18 third-party domains that are not used in the corresponding landing pages (Fig. 8b). Also, per Fig. 8b, for 10% of the web sites in $\mathcal{H}_{1K}$, internal pages fetch content from 80 or more third-party domains that are not observed on the landing pages.

Implications for prior work. Ikram et al. fetched the landing pages of web sites in Alexa Top 200K and built dependency chains of resources for each web site [55]. They used an antivirus aggregator tool and reported that 1.2% of the third-party domains are malicious. Similarly, Yue et al. measured the amount of third-party JavaScript on web sites using a data set comprising 6805 landing pages [113]. We find that collectively, internal pages of a web site fetch content from a much larger number of third-parties than landing pages. The aforementioned prior work, hence, would underestimate the dependency structure for a web site as a whole. Urban et al. have recently studied the differences in third-party dependencies of landing and internal pages, and their results largely agree with ours [101]. There are also a few other studies that address similar problems but include internal pages in their analyses [62, 78, 97].

6.3 Ads and Trackers

While advertisements help content providers monetize content, the pervasive tracking, monitoring, and profiling of users purportedly for targeted ad placements and customization has faced strong criticism from both industry and end users [32, 33, 75]. The advent of GDPR, furthermore, has led to significant changes in the use of trackers as well as end-user-data collection practices. With significant differences in content across landing and internal pages, we simply ask, in this section, if the use (or the lack thereof) of ads and trackers is consistent across the two types of pages.

To detect advertisement and tracking related requests, we used the Brave Browser Adblock library [20] coupled with Easylist [37]. Easylist is a list of over 73,000 URL patterns associated with online ads and tracking. Popular ad blockers, such as AdBlock Plus, and uBlock Origin use this list. We counted all HTTP requests on a web page that would have been blocked by Brave Adblock. We label all such blocked requests as those corresponding to trackers for ease of exposition. The CDFs of the number of trackers per page (in Fig. 8c) reveal that at the 80^{th} percentile, internal pages have 20 tracking requests while landing pages have 28. Furthermore, in about 10% of the web sites in $\mathcal{H}_{1K}$, internal pages have no trackers while the corresponding landing pages do.

Header bidding (HB) is a new online advertising technology that initiates ad auctions from the browser instead of relying on an ad server (see [11] for detail). We used the open source tools from [11] to analyze HB ads in $\mathcal{H}_{t100}$ and $\mathcal{H}_{b100}$. Out of the 200 web sites, we find that 17 have HB ads on landing pages. An additional 12 web sites have such ads on internal pages, but not on the landing page. We also find that among web sites that have HB ads, internal pages have 7 ad slots in the 80^{th} percentile whereas landing pages have 9. The differing use of advertisements and trackers between landing and internal pages has implications especially for studies measuring compliance to GDPR and privacy leakage.

Implications for prior work. Pachilakis et al. crawled the landing pages of the top 35K web sites, and detected 5000 that have header bidding ads [81]. Then, they crawled these web sites daily for 34 days and report on the number of ad slots and many other metrics. Owing to the exclusion of internal pages, this study will miss web sites that have HB ads only on internal pages. Lerner et al. present a seminal study of user tracking from 1996 to 2016 [65]. They use the Wayback Machine [56] to crawl the Alexa Top 500 web sites, and report extensive historical trends on the trackers, techniques, and prevalence found on these web sites. Based on our observations, this study will overestimate tracker activity on the web, since it considers only landing pages.

7 ON SELECTING INTERNAL PAGES

Selecting a "representative" set of internal pages from a web site without any traffic measurements is hard. In addressing this problem and creating *Hispar*, we used search engine results. Retrieving search results using a search-engine API, however, is not free. Google (Bing) charges, for instance, $5 ($3) per 1000 queries [47, 72], although Bing is effectively cheaper because it returns more results per query. However, we encountered some bugs in Bing's API and instead opted for Google. Generating a list of 100,000 URLs using Google would require at least 10,000 queries, and under standard pricing, would cost $50. Many queries return, however, less than 10 unique URLs; hence our cost has consistently been around $70 per list. About half of the studies that received a "major" revision score in our review (§2) used 500 or fewer web sites. Including up to 50 internal pages per web site for these studies would cost less than $20. We provide, nevertheless, the weekly $\mathcal{H}_{2K}$ lists of 100,000 URLs free of charge for the community's convenience, and to spur future web-perf. studies to include internal pages. We now discuss other approaches to selecting internal pages.

Involve publishers. Web page performance is of paramount importance to publishers since poor performance often translates to a significant loss in revenue [2, 22]. Hence, publishers are most likely to be interested in determining whether a given optimization (e.g., using a specific CDN to reduce page-load times) is *representative*, i.e., whether it generalizes to the majority of the site's pages. Publishers can either select a set of internal pages or even construct a set of *synthetic* internal pages that serve as a good representation of the internal pages on their site. These samples can be published at a *Well-Known* URI [79] or through extension of mechanisms like `robots.txt` [60]. This approach is similar in purpose to web-standards compliance and browser-performance benchmarks (e.g., Acid3 [98], Kraken [57], and JetStream2 [21]), albeit implemented in a more distributed way: Each publisher specifies a benchmark—the representative internal page(s)—that we *must* use for any performance measurement or optimization on their web site. Ensuring that this set of internal pages does not go stale as the site contents change, however, is a challenging task.

Nudge web-browser vendors. Major web-browser vendors such as Mozilla and Google already collect anonymized user data (e.g., URLs of web pages visited by the end users) via projects such as Telemetry [74] and CrUX [45], respectively. Such data can also be gathered from Google Analytics and other user-tracking platforms that a majority of the Alexa Top 1M web sites already use [24]. Web-perf. efforts will immensely benefit if such projects make their anonymized data sets publicly available. Indeed, Alexa and other top-list providers may leverage their existing vantage points to help select internal pages in different web sites. Although web content may significantly differ based on query parameters, session data, cookies, location, and time, publishing at least the most popular URLs from such data sets will be the first step towards improving web-perf. measurement. Sharing this aggregated data does not violate the privacy of end users.

Learn web page characteristics. We could also use machine-learning tools to learn the structure and characteristics of different web pages. Augmented with other parameters such as the end-user's bandwidth and last-mile latency, the learned model can help in evaluating how a given optimization will perform under various scenarios, each representing a page with a distinct set of characteristics.

8 RELATED WORK

There is an extensive body of literature on web performance measurement and optimization, dating all the way back to when the web came into existence [43]. Several useful guidelines on conducting sound Internet measurements have also been published [7, 8, 36, 83]. Nearly every aspect of web-perf. studies has also faced extensive scrutiny [16, 23, 38, 46, 59, 67, 77, 84, 93]. Our work is orthogonal to these studies and shines light on a heretofore neglected aspect: the exclusion of internal pages in web-perf. studies.

Recently, Scheitle et al. showed that the choice of top lists such as Alexa and Quantcast has implications for web-perf. studies because of a lower than expected overlap among these lists [93]; they also show that there is considerable churn within a given list. Pochat et al. provide a new top list that averages four different lists over 30 days [84]. However, these efforts do not address the problem that such top lists can only be directly used for landing pages.

Kaizer and Gupta's work on how the privacy-related behavior of web sites differs based on whether the user is logged in or not is most similar to our work. [58] To the best of our knowledge, our work is the first one to highlight the intrinsic differences between landing and internal pages of web sites and their impact on past research.

9 CONCLUSION

We compared the landing page with a set of internal pages for 1000 web sites, constituting *Hispar* $\mathcal{H}_{1K}$, and discovered that landing pages differ substantially from internal pages. These differences have implications for virtually all web-perf. studies that examined simply the landing pages. We manually reviewed more than a hundred different web-perf. studies published at top-tier conferences and found that about two-thirds of the studies would have to be revised if they were to apply to internal pages.

We released our measurements data set and analyses scripts for use of other researchers. We also made *Hispar*, including all its weekly updates, and tools required for customizing or regenerating the list publicly available [50].

Armed with the insights and tools from this study, we hope that future work on web performance measurement and optimization will start including the larger, slower, and neglected part of the web for more sound measurements, optimizations, and evaluations. While we provide several suggestion on selecting internal pages, we hope that our study paves the way for a discussion concerning the selection and curation of internal pages, and eliciting support from content providers, search engines, and browser vendors to generate a rich and scalable version of *Hispar*, or an even better alternative.

ACKNOWLEDGEMENTS

We thank Matteo Varvello, our shepherd, and the anonymous reviewers for their insightful comments.

This work is supported in part by the National Science Foundation under Award No. 1763742 and 1901047.

REFERENCES

[1] Bernhard Ager, Wolfgang Mühlbauer, Georgios Smaragdakis, and Steve Uhlig. 2010. Comparing DNS Resolvers in the Wild. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement (IMC '10)*. Association for Computing Machinery, New York, NY, USA, 15–21.

[2] Akamai. 2015. Akamai "10For10". https://www.akamai.com/us/en/multimedia/documents/brochure/akamai-10for10-brochure.pdf. (July 2015).

[3] Akamai. 2018. Using Akamai Pragma headers to investigate or troubleshoot Akamai content delivery. https://community.akamai.com/customers/s/article/Using-Akamai-Pragma-headers-to-investigate-or-troubleshoot-Akamai-content-delivery. (2018). [Last accessed on May 12, 2020].

[4] Rami Al-Dalky and Michael Rabinovich. 2020. Revisiting Comparative Performance of DNS Resolvers in the IPv6 and ECS Era. (2020). arXiv:cs.NI/2007.00651

[5] alexa.com. 2019. Alexa—Top sites. https://www.alexa.com/topsites. (2019). [Last accessed on November 30, 2019].

[6] alexa.com. 2019. Alexa | About Us. https://www.alexa.com/about. (2019). [Last accessed on December 13, 2019].

[7] Mark Allman. 2013. On Changing the Culture of Empirical Internet Assessment. *SIGCOMM Comput. Commun. Rev.* 43, 3 (July 2013), 78–83.

[8] Mark Allman, Robert Beverly, and Brian Trammell. 2017. Principles for Measurability in Protocol Design. *SIGCOMM Comput. Commun. Rev.* 47, 2 (May 2017), 2–12.

[9] Mshabab Alrizah, Sencun Zhu, Xinyu Xing, and Gang Wang. 2019. Errors, Misunderstandings, and Attacks: Analyzing the Crowdsourcing Process of Ad-Blocking Systems. In *Proceedings of the Internet Measurement Conference (IMC '19)*. Association for Computing Machinery, New York, NY, USA, 230–244.

[10] Johanna Amann, Oliver Gasser, Quirin Scheitle, Lexi Brent, Georg Carle, and Ralph Holz. 2017. Mission Accomplished?: HTTPS Security After Diginotar. In *Proceedings of the 2017 Internet Measurement Conference (IMC '17)*. ACM, New York, NY, USA, 325–340.

[11] Waqar Aqeel, Debopam Bhattacherjee, Balakrishnan Chandrasekaran, P. Brighten Godfrey, Gregory Laughlin, Bruce Maggs, and Ankit Singla. 2020. Untangling Header Bidding Lore: Some myths, some truths, and some hope. In *Passive and Active Measurement*, Anna Sperotto, Alberto Dainotti, and Burkhard Stiller (Eds.). Springer International Publishing, Cham, 280–297.

[12] Muhammad Ahmad Bashir, Sajjad Arshad, Engin Kirda, William Robertson, and Christo Wilson. 2018. How Tracking Companies Circumvented Ad Blockers Using WebSockets. In *Proceedings of the Internet Measurement Conference 2018 (IMC '18)*. ACM, New York, NY, USA, 471–477.

[13] Mike Belshe, Roberto Peon, and Martin Thomson. 2015. Hypertext Transfer Protocol Version 2 (HTTP/2). RFC 7540. (May 2015).

[14] Daniel S. Berger, Ramesh K. Sitaraman, and Mor Harchol-Balter. 2017. AdaptSize: Orchestrating the Hot Object Memory Cache in a Content Delivery Network. In *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)*. USENIX Association, Boston, MA, 483–498.

[15] Debopam Bhattacherjee, Waqar Aqeel, Ilker Nadi Bozkurt, Anthony Aguirre, Balakrishnan Chandrasekaran, P. Brighten Godfrey, Gregory Laughlin, Bruce Maggs, and Ankit Singla. 2018. Gearing Up for the 21st Century Space Race. In *Proceedings of the 17th ACM Workshop on Hot Topics in Networks (HotNets '18)*. ACM, New York, NY, USA, 113–119.

[16] Enrico Bocchi, Luca De Cicco, and Dario Rossi. 2016. Measuring the Quality of Experience of Web Users. In *Proceedings of the 2016 Workshop on QoE-based Analysis and Management of Data Communication Networks (Internet-QoE '16)*. ACM, New York, NY, USA, 37–42.

[17] Timm Böttger, Felix Cuadrado, Gianni Antichi, Eder Leão Fernandes, Gareth Tyson, Ignacio Castro, and Steve Uhlig. 2019. An Empirical Study of the Cost of DNS-over-HTTPS. In *Proceedings of the Internet Measurement Conference (IMC '19)*. Association for Computing Machinery, New York, NY, USA, 15–21.

[18] Ilker Nadi Bozkurt, Anthony Aguirre, Balakrishnan Chandrasekaran, P. Brighten Godfrey, Gregory Laughlin, Bruce Maggs, and Ankit Singla. 2017. Why Is the Internet so Slow?!. In *Passive and Active Measurement: 18th International Conference, PAM 2017, Sydney, NSW, Australia, March 30-31, 2017, Proceedings*, Mohamed Ali Kaafar, Steve Uhlig, and Johanna Amann (Eds.). Springer International Publishing, Cham, 173–187.

[19] BrightEdge. 2019. BrightEdge Channel Report. https://www.brightedge.com/resources/research-reports/channel_share. (2019).

[20] Brave Browser. 2019. Brave Ad Block. https://github.com/brave/ad-block. (2019). [Last accessed on January 25, 2020].

[21] Browser Benchmarks. 2019. JetStream2. https://browserbench.org/JetStream/in-depth.html. (2019).

[22] Jake Brutlag. 2009. Speed Matters for Google Web Search. http://goo.gl/t7qGN8. (June 2009).

[23] Jake Brutlag, Zoe Abrams, and Pat Meenan. 2011. Above the Fold Time: Measuring Web Page Performance Visually. (March 2011). https://conferences.oreilly.com/velocity/velocity-mar2011/public/schedule/detail/18692.

[24] Builtwith. 2020. Google Analytics Usage Statistics. https://trends.builtwith.com/analytics/Google-Analytics. (2020).

[25] Michael Butkiewicz, Harsha V. Madhyastha, and Vyas Sekar. 2011. Understanding Website Complexity: Measurements, Metrics, and Implications. In *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference (IMC '11)*. ACM, New York, NY, USA, 313–328.

[26] Michael Butkiewicz, Daimeng Wang, Zhe Wu, Harsha V. Madhyastha, and Vyas Sekar. 2015. Klotski: Reprioritizing Web Content to Improve User Experience on Mobile Devices. In *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15)*. USENIX Association, Oakland, CA, 439–453.

[27] Stefano Calzavara, Alvise Rabitti, and Michele Bugliesi. 2016. Content Security Problems? Evaluating the Effectiveness of Content Security Policy in the Wild. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16)*. Association for Computing Machinery, New York, NY, USA, 1365–1375.

[28] Andrea Cardaci. 2019. Chrome Page Graph. https://github.com/cyrus-and/chrome-page-graph/. (2019). [Last accessed on January 25, 2020].

[29] Chrome. 2020. Chrome DevTools Protocol. https://chromedevtools.github.io/devtools-protocol/tot/Network/. (2020). [Last accessed on January 25, 2020].

[30] Chromium Blog. 2015. A QUIC update on Google's experimental transport. https://blog.chromium.org/2015/04/a-quic-update-on-googles-experimental.html. (2015).

[31] Cisco. 2019. Visual Networking Index: Forecast and Trends, 2017-2022 White Paper. https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html. (February 2019).

[32] Nicholas Confessore. 2018. Cambridge Analytica and Facebook: The Scandal and the Fallout So Far. https://www.nytimes.com/2018/04/04/us/politics/cambridge-analytica-scandal-fallout.html. (April 2018).

[33] Bennet Cyphers and Gennie Gebhart. 2019. *Behind the One-Way Mirror: A Deep Dive Into the Technology of Corporate Surveillance.* Technical Report. Electronic Frontier Foundation.

[34] Diego Neves da Hora, Alemnew Sheferaw Asrese, Vassilis Christophides, Renata Teixeira, and Dario Rossi. 2018. Narrowing the Gap Between QoS Metrics and Web QoE Using Above-the-fold Metrics. In *Passive and Active Measurement*, Robert Beverly, Georgios Smaragdakis, and Anja Feldmann (Eds.). Springer International Publishing, Cham, 31–43.

[35] Zakir Durumeric, James Kasten, Michael Bailey, and J. Alex Halderman. 2013. Analysis of the HTTPS Certificate Ecosystem. In *Proceedings of the 2013 Conference on Internet Measurement Conference (IMC '13)*. ACM, New York, NY, USA, 291–304.

[36] Zakir Durumeric, Eric Wustrow, and J. Alex Halderman. 2013. ZMap: Fast Internet-wide Scanning and Its Security Applications. In *Presented as part of the 22nd USENIX Security Symposium (USENIX Security 13)*. USENIX, Washington, D.C., 605–620.

[37] EasyList. 2019. Easylist filter list project. https://easylist.to. (August 2019).

[38] Theresa Enghardt, Thomas Zinner, and Anja Feldmann. 2019. Web Performance Pitfalls. In *Passive and Active Measurement*, David Choffnes and Marinho Barcellos (Eds.). Springer International Publishing, Cham, 286–303.

[39] Fastly. 2019. Understanding cache HIT and MISS headers with shielded services. https://docs.fastly.com/en/guides/understanding-cache-hit-and-miss-headers-with-shielded-services. (2019). [Last accessed on May 12, 2020].

[40] Adrienne Porter Felt, Richard Barnes, April King, Chris Palmer, Chris Bentzel, and Parisa Tabriz. 2017. Measuring HTTPS Adoption on the Web. In *26th USENIX Security Symposium (USENIX Security 17)*. USENIX Association, Vancouver, BC, 1323–1338.

[41] Rand Fishkin and Thomas Høgenhaven. 2013. *Inbound marketing and SEO: Insights from the Moz Blog.* John Wiley & Sons, Hoboken, NJ, USA.

[42] Oliver Gasser, Benjamin Hof, Max Helm, Maciej Korczyński, Ralph Holz, and Georg Carle. 2018. In Log We Trust: Revealing Poor Security Practices with Certificate Transparency Logs and Internet Measurements. In *Passive and Active Network Measurement: 19th International Conference, PAM 2018, Berlin, Germany, ,March 26-27, 2018. Proceedings*, Robert Beverly, Georgios Smaragdakis, and Anja Feldmann (Eds.). Springer International Publishing, Cham, 173–18.

[43] Steven Glassman. 1994. A Caching Relay for the World Wide Web. In *Selected Papers of the First Conference on World-Wide Web*. Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, 165–173.

[44] Google. 2019. About PageSpeed Insights. https://developers.google.com/speed/docs/insights/v5/about. (2019).

[45] Google. 2019. Chrome User Experience Report. https://developers.google.com/web/tools/chrome-user-experience-report. (2019).

[46] Google. 2019. Speed Index. https://web.dev/speed-index/. (2019).

[47] Google. 2020. Google Custom Search. https://developers.google.com/custom-search/docs/overview. (2020). [Last accessed on April 26, 2020].

[48] Google. 2020. HTTPS encryption on the web. https://transparencyreport.google.com/https/overview. (2020).

[49] Google Public DNS. 2019. Performance Benefits. https://developers.google.com/speed/public-dns/docs/performance. (October 2019).

[50] Hispar Project. 2020. Pagetypes data set, Hispar list tools and archives. https://hispar.cs.duke.edu/. (2020).

[51] Jeff Hodges, Collin Jackson, and Adam Barth. 2012. HTTP Strict Transport Security (HSTS). RFC 6797. (Nov. 2012).

[52] Paul E. Hoffman and Patrick McManus. 2018. DNS Queries over HTTPS (DoH). RFC 8484. (Oct. 2018).

[53] HTTP Archive. 2019. State of JavaScript. https://httparchive.org/reports/state-of-javascript#bytesJs. (2019). [Last accessed on December 5, 2019].

[54] Dan Hubbard. 2016. Cisco Umbrella 1 Million. https://umbrella.cisco.com/blog/2016/12/14/cisco-umbrella-1-million/. (December 2016).

[55] Muhammad Ikram, Rahat Masood, Gareth Tyson, Mohamed Ali Kaafar, Noha Loizon, and Roya Ensafi. 2019. The Chain of Implicit Trust: An Analysis of the Web Third-Party Resources Loading. In *The World Wide Web Conference (WWW '19)*. Association for Computing Machinery, New York, NY, USA, 2851–2857.

[56] Internet Archive. 2019. Wayback Machine. https://archive.org. (2019). [Last accessed on December 5, 2019].

[57] Erica Jostedt. 2010. Release the Kraken. https://blog.mozilla.org/blog/2010/09/14/release-the-kraken-2/. (2010).

[58] Andrew J. Kaizer and Minaxi Gupta. 2016. Characterizing Website Behaviors Across Logged-in and Not-Logged-in Users. In *Proceedings of the 2016 Internet Measurement Conference (IMC '16)*. Association for Computing Machinery, New York, NY, USA, 111–117.

[59] Conor Kelton, Jihoon Ryoo, Aruna Balasubramanian, and Samir R. Das. 2017. Improving User Perceived Page Load Times Using Gaze. In *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)*. USENIX Association, Boston, MA, 545–559.

[60] Martijn Koster. 1994. A Standard for Robot Exclusion. https://www.robotstxt.org/orig.html. (1994).

[61] Michael Kranch and Joseph Bonneau. 2015. Upgrading HTTPS in Mid-Air: An Empirical Study of Strict Transport Security and Key Pinning. (February 2015).

[62] Deepak Kumar, Zane Ma, Zakir Durumeric, Ariana Mirian, Joshua Mason, J. Alex Halderman, and Michael Bailey. 2017. Security Challenges in an Increasingly Tangled Web. In *Proceedings of the 26th International Conference on World Wide Web (WWW '17)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 677–684.

[63] Adam Langley, Alistair Riddoch, Alyssa Wilk, Antonio Vicente, Charles Krasic, Dan Zhang, Fan Yang, Fedor Kouranov, Ian Swett, Janardhan Iyengar, Jeff Bailey, Jeremy Dorfman, Jim Roskind, Joanna Kulik, Patrik Westin, Raman Tenneti, Robbie Shade, Ryan Hamilton, Victor Vasiliev, Wan-Teh Chang, and Zhongyi Shi. 2017. The QUIC Transport Protocol: Design and Internet-Scale Deployment. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication (SIGCOMM '17)*. ACM, New York, NY, USA, 183–196.

[64] Ben Laurie, Adam Langley, and Emilia Kasper. 2013. Certificate Transparency. RFC 6962. (June 2013).

[65] Adam Lerner, Anna Kornfeld Simpson, Tadayoshi Kohno, and Franziska Roesner. 2016. Internet Jones and the Raiders of the Lost Trackers: An Archaeological Study of Web Tracking from 1996 to 2016. In *25th USENIX Security Symposium (USENIX Security 16)*. USENIX Association, Austin, TX, 997–1014.

[66] Zhichun Li, Ming Zhang, Zhaosheng Zhu, Yan Chen, Albert Greenberg, and Yi-Min Wang. 2010. WebProphet: Automating Performance Prediction for Web Services. In *7th USENIX Symposium on Networked Systems Design and Implementation (NSDI 10)*. USENIX Association, San Jose, CA, 143–158.

[67] Bruce W.N. Lo and Rosy Sharma Sedhain. 2006. How Reliable are Website Rankings? Implications for E-Business Advertising and Search. *Issues in Information Systems* VII, 2 (2006), 233–238.

[68] majestic.com. 2019. Majestic Help Centre — Frequently Asked Questions. https://majestic.com/help/faq. (2019). [Last accessed on November 30, 2019].

[69] majestic.com. 2019. The Majestic Million. https://majestic.com/reports/majestic-million. (2019). [Last accessed on November 30, 2019].

[70] Kieran McDonald. 2012. Minimizing Answer Defects. https://blogs.bing.com/search-quality-insights/2012/08/20/minimizing-answer-defects. (2012). [Last accessed on January 25, 2020].

[71] MDN web docs. 2019. Cacheable. https://developer.mozilla.org/en-US/docs/Glossary/cacheable. (March 2019).

[72] Microsoft Azure. 2019. Bing Web Search. https://azure.microsoft.com/en-us/services/cognitive-services/bing-web-search-api/. (2019). [Last accessed on December 5, 2019].

[73] Giovane C. M. Moura, John Heidemann, Ricardo de O. Schmidt, and Wes Hardaker. 2019. Cache Me If You Can: Effects of DNS Time-to-Live (extended). In *Proceedings of the ACM Internet Measurement Conference*. ACM, Amsterdam, the Netherlands, 101–115.

[74] Mozilla. 2017. Telemetry. https://wiki.mozilla.org/Telemetry. (2017).

[75] Mozilla Firefox. 2019. Content blocking. https://support.mozilla.org/en-US/kb/content-blocking. (2019). [Last accessed on October 16, 2019].

[76] Ravi Netravali, Ameesh Goyal, James Mickens, and Hari Balakrishnan. 2016. Polaris: Faster Page Loads Using Fine-grained Dependency Tracking. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*. USENIX Association, Santa Clara, CA, 123–136.

[77] 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.

[78] Nick Nikiforakis, Luca Invernizzi, Alexandros Kapravelos, Steven Van Acker, Wouter Joosen, Christopher Kruegel, Frank Piessens, and Giovanni Vigna. 2012. You Are What You Include: Large-Scale Evaluation of Remote Javascript Inclusions. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security (CCS '12)*. Association for Computing Machinery, New York, NY, USA, 736–747.

[79] Mark Nottingham. 2019. Well-Known Uniform Resource Identifiers (URIs). RFC 8615. (May 2019).

[80] Addy Osmani. 2017. Preload, Prefetch And Priorities in Chrome. https://medium.com/reloading/preload-prefetch-and-priorities-in-chrome-776165961bbf. (2017). [Last accessed on January 25, 2020].

[81] Michalis Pachilakis, Panagiotis Papadopoulos, Evangelos P. Markatos, and Nicolas Kourtellis. 2019. No More Chasing Waterfalls: A Measurement Study of the Header Bidding Ad-Ecosystem. In *Proceedings of the 2017 Internet Measurement Conference (IMC '19)*. ACM, New York, NY, USA, 280–293.

[82] Muhammad Talha Paracha, Balakrishnan Chandrasekaran, David Choffnes, and Dave Levin. 2020. A Deeper Look at Web Content Availability and Consistency over HTTP/S. (2020).

[83] Vern Paxson. 2004. Strategies for Sound Internet Measurement. In *Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement (IMC '04)*. Association for Computing Machinery, New York, NY, USA, 263–271.

[84] Victor Le Pochat, Tom Van Goethem, Samaneh Tajalizadehkhoob, Maciej Korczyński, and Wouter Joosen. 2019. Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation. (2019).

[85] John W. Pratt and Jean D. Gibbons. 1981. *Kolmogorov-Smirnov Two-Sample Tests.* Springer New York, New York, NY, 318–344. https://doi.org/10.1007/978-1-4612-5931-2_7
[86] Shankaranarayanan Puzhavakath Narayanan, Yun Seong Nam, Ashiwan Sivakumar, Balakrishnan Chandrasekaran, Bruce Maggs, and Sanjay Rao. 2016. Reducing Latency Through Page-aware Management of Web Objects by Content Delivery Networks. In *Proceedings of the 2016 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Science (SIGMETRICS '16).* ACM, New York, NY, USA, 89–100.
[87] Quantcast. 2020. The World's largest Audience Behavior Platform for the Open Internet. https://www.quantcast.com/about-us/. (2020). Last accessed on January 4, 2020.
[88] quantcast.com. 2019. Quantcast—Top sites. https://www.quantcast.com/top-sites/. (2019). [Last accessed on November 30, 2019].
[89] Sivasankar Radhakrishnan, Yuchung Cheng, Jerry Chu, Arvind Jain, and Barath Raghavan. 2011. TCP Fast Open. In *Proceedings of the Seventh COnference on emerging Networking EXperiments and Technologies (CoNEXT '11).* ACM, New York, NY, USA, 21:1–21:12.
[90] Eric Rescorla. 2018. The Transport Layer Security (TLS) Protocol Version 1.3. RFC 8446. (Aug. 2018).
[91] Ian Rogers. 2002. Understanding Google Page Rank. http://ianrogers.uk/google-page-rank/. (Aug. 2002).
[92] Vaspol Ruamviboonsuk, Ravi Netravali, Muhammed Uluyol, and Harsha V. Madhyastha. 2017. Vroom: Accelerating the Mobile Web with Server-Aided Dependency Resolution. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication (SIGCOMM '17).* Association for Computing Machinery, New York, NY, USA, 390–403.
[93] Quirin Scheitle, Oliver Hohlfeld, Julien Gamba, Jonas Jelten, Torsten Zimmermann, Stephen D. Strowes, and Narseo Vallina-Rodriguez. 2018. A Long Way to the Top: Significance, Structure, and Stability of Internet Top Lists. In *Proceedings of the Internet Measurement Conference 2018 (IMC '18).* ACM, New York, NY, USA, 478–493.
[94] Justine Sherry, Chang Lan, Raluca Ada Popa, and Sylvia Ratnasamy. 2015. Blind-Box: Deep Packet Inspection over Encrypted Traffic. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication (SIGCOMM '15).* Association for Computing Machinery, New York, NY, USA, 213–226.
[95] Ankit Singla, Balakrishnan Chandrasekaran, P. Brighten Godfrey, and Bruce Maggs. 2014. The Internet at the Speed of Light. In *Proceedings of the 13th ACM Workshop on Hot Topics in Networks (HotNets-XIII).* ACM, New York, NY, USA, 1:1–1:7.
[96] Ramesh K. Sitaraman, Mangesh Kasbekar, Woody Lichtenstein, and Manish Jain. 2014. *Overlay Networks: An Akamai Perspective.* John Wiley & Sons, Inc., Hoboken, NJ, USA, Chapter 16, 305–328.
[97] Ben Stock, Martin Johns, Marius Steffens, and Michael Backes. 2017. How the Web Tangled Itself: Uncovering the History of Client-Side Web (In)Security. In *26th USENIX Security Symposium (USENIX Security 17).* USENIX Association, Vancouver, BC, 971–987.
[98] The Web Standards Project. 2008. Acid3 Browser Test. https://www.webstandards.org/action/acid3/. (2008).
[99] TurboBytes. 2019. cdnfinder. https://github.com/turbobytes/cdnfinder. (2019). [Last accessed on December 5, 2019].
[100] umbrella.com. 2019. Umbrella Popularity List—Top Million Domains. https://docs.umbrella.com/investigate-api/docs/top-million-domains. (2019). [Last accessed on November 30, 2019].
[101] Tobias Urban, Martin Degeling, Thorsten Holz, and Norbert Pohlmann. 2020. Beyond the Front Page:Measuring Third Party Dynamics in the Field. In *Proceedings of The Web Conference 2020 (WWW '20).* Association for Computing Machinery, New York, NY, USA, 1275–1286.
[102] Benjamin VanderSloot, Johanna Amann, Matthew Bernhard, Zakir Durumeric, Michael Bailey, and J. Alex Halderman. 2016. Towards a Complete View of the Certificate Ecosystem. In *Proceedings of the 2016 Internet Measurement Conference (IMC '16).* ACM, New York, NY, USA, 543–549.
[103] Jamshed Vesuna, Colin Scott, Michael Buettner, Michael Piatek, Arvind Krishnamurthy, and Scott Shenker. 2016. Caching Doesn't Improve Mobile Web Performance (Much). In *2016 USENIX Annual Technical Conference (USENIX ATC 16).* USENIX Association, Denver, CO, 159–165.
[104] W3C. 2012. HTTP Archive (HAR) format. https://w3c.github.io/web-performance/specs/HAR/Overview.html. (August 2012).
[105] W3C. 2012. Navigation Timing. https://www.w3.org/TR/navigation-timing/. (December 2012).
[106] W3C. 2012. WebIntents/MIME Types. https://www.w3.org/wiki/WebIntents/MIME_Types. (September 2012).
[107] W3C. 2019. Resource Hints. https://www.w3.org/TR/resource-hints/. (December 2019).
[108] Xiao Sophia Wang, Aruna Balasubramanian, Arvind Krishnamurthy, and David Wetherall. 2013. Demystifying Page Load Performance with WProf. In *Presented as part of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13).* USENIX, Lombard, IL, 473–485.
[109] Xiao Sophia Wang, Arvind Krishnamurthy, and David Wetherall. 2016. Speeding up Web Page Loads with Shandian. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16).* USENIX Association, Santa Clara, CA, 109–122.
[110] WebPagetest. 2012. Speed Index - WebPagetest Documentation. https://sites.google.com/a/webpagetest.org/docs/using-webpagetest/metrics/speed-index. (2012). [Last accessed on December 4, 2019].
[111] Maarten Wijnants, Robin Marx, Peter Quax, and Wim Lamotte. 2018. HTTP/2 Prioritization and Its Impact on Web Performance. In *Proceedings of the 2018 World Wide Web Conference (WWW '18).* International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 1755–1764.
[112] Konrad Wolsing, Jan Rüth, Klaus Wehrle, and Oliver Hohlfeld. 2019. A Performance Perspective on Web Optimized Protocol Stacks: TCP+TLS+HTTP/2 vs. QUIC. In *Proceedings of the Applied Networking Research Workshop (ANRW '19).* Association for Computing Machinery, New York, NY, USA, 1–7.
[113] Chuan Yue and Haining Wang. 2009. Characterizing Insecure Javascript Practices on the Web. In *Proceedings of the 18th International Conference on World Wide Web (WWW '09).* Association for Computing Machinery, New York, NY, USA, 961–970.

A TRENDS IN DIFFERENCES

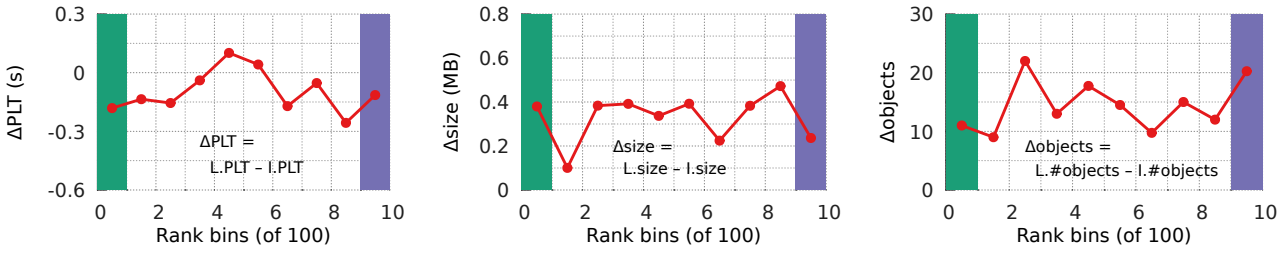
In this appendix, we highlight how some of the differences between landing and internal pages that we discussed earlier (in §4 and §5) vary based on the popularity ranking or (Alexa) category of web sites.

To investigate how page-type differences vary for a certain metric μ (e.g., page size, number of domains contacted), we calculated the difference ($\Delta\mu$) between the value of μ for the landing page and the median value of μ for that web site's internal pages. We then divided the web sites in $\mathcal{H}_{1K}$ into bins of size 100 each based on their popularity ranks in $\mathcal{A}_{1M}$. Last, we calculated the median values of $\Delta\mu$ for each rank bin and plotted them as a function of the rank bins.

We see a trend reversal in page-load time difference over popularity ranks (Fig. 9a): ΔPLT is negative for most rank bins, i.e., landing pages are generally faster than internal pages. For web sites with ranks between 400 and 600, ΔPLT is, however, positive (up to 100 ms), implying that landing pages of these web sites are slower than their internal pages. Although there is no reversal in trends for page size (Fig. 9b) and object count (Fig. 9c), the magnitude of difference varies significantly based on popularity ranks.

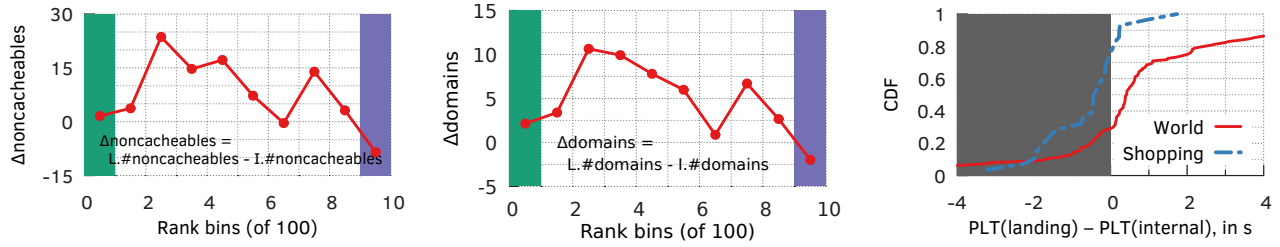
We also observe trend reversals for the number of non-cacheable objects (Fig. 10a) and unique domain names (Fig. 10b) found on landing versus internal pages. For web sites between ranks 200 and 300, landing pages have 24 more non-cacheable objects (and 11 more unique domain names) than the corresponding internal pages. Between ranks 900 and 1000, however, landing pages have 8 fewer non-cacheable objects (and 2 fewer unique domain names) than the corresponding internal pages.

Last, we discover a category-based reversal in page-load time (PLT): For almost all Alexa top-level categories of web sites, the majority of landing pages are faster than the corresponding median internal page, but for the *World* category, the majority of landing pages are slower. Fig. 10c plots the difference between the landing-page PLT and median of internal-page PLTs for each web site in two categories, *World* and *Shopping*. The *Shopping* category follows the general trend (as in Fig. 2c): for about 77% of web sites, landing pages are faster than internal pages. The *World* category bucks the trend: For about 70% of the web sites, landing pages are slower than internal pages. The *World* category contains web sites that are popular internationally, but not in the U.S. (e.g. `baidu.com`).



(a) (b) (c)

Figure 9: ***Differences between landing (L) and internal (I) pages vary significantly based on popularity ranks for (a) page-load times, (b) total page size, and (c) number of objects. Top 100 and bottom 100 web sites by popularity rank are highlighted.***



(a) (b) (c)

Figure 10: ***Trend reversals in differences between landing (L) and internal (I) pages for (a) non-cacheable objects and (b) unique domain names. (c) Landing pages in World category are generally slower than internal pages, whereas they are faster in Shopping.***

Objects in these web sites would probably not result in cache "hits" at the CDN if fetched from the U.S. as we did. These web sites may not even be served from a CDN site in North America.

The differences between landing and internal pages are not uniform, but vary significantly based on popularity ranking of the concerned web site. The number of objects on landing pages, for instance, is not simply a constant multiple of the objects on internal pages. Hence, researchers should actually measure internal pages (in addition to landing pages) for their insights to generalize to a whole web site.