Understanding and Mitigating Webpage Data Bloat: Causes and Preventive Measures

Aleksi Saarinen
Aalto University
Espoo, Finland

Muhammad Zubair Farooqi
Aalto University
Espoo, Finland

Matti Pärssinen
Aalto University
Espoo, Finland

Jukka Manner
Aalto University
Espoo, Finland

ABSTRACT

The World Wide Web and web pages play an integral role, serving as the backbone for the myriad of services we rely on regularly. The size of web pages has increased constantly, which is not optimal considering the growing awareness of the substantial energy and resource consumption associated with Information and Communication Technology (ICT). This paper presents an analysis of 10,000 popular web pages and studies ways to use less data while still retaining the essential content. Our comprehensive analysis reveals the potential for remarkable data savings, amounting to over half of the data currently utilized. This paper highlights a conscientious approach to web page construction towards more environmentally sustainable digital services.

KEYWORDS

World Wide Web, Web services, Performance evaluation, Image compression, Sustainable ICT

ACM Reference Format:
Aleksi Saarinen, Muhammad Zubair Farooqi, Matti Pärssinen, and Jukka Manner. 2024. Understanding and Mitigating Webpage Data Bloat: Causes and Preventive Measures. In *Proceedings of 3rd Workshop on Sustainable Computer Systems (HotCarbon'24)*. Santa Cruz, CA, 7 pages.

1 INTRODUCTION

The Internet is the platform for a huge range of services. At the heart of most services is the World Wide Web and its web pages. Web pages have evolved from simple static pages with text-based information to pages that provide different types of services, such as banking, social networking, video on demand and online shopping. These services are often complex and thus require more resources and use more data than the simple text-based pages.

Predictions from various studies indicate that if current trends continue without significant intervention, ICT energy consumption could constitute a much larger percentage of global electricity use, with some estimates projecting it could rise to 20% by 2030 [1]. This underscores the urgency for immediate and sustained action to enhance energy efficiency across all ICT domains, from individual devices to large-scale data centers. Andrae and Edler [2] also predict that the ICT share of the global electricity usage in 2030 could reach 51% in the worst case scenario and 8% in the best scenario, with the expectation being between these predictions at 21%. In 2012, this share was estimated to be only 4.7%. Especially data center and fixed wired network energy usage is predicted to increase significantly, while the consumer device energy usage should decrease. The share of the networking area is also growing rapidly, partly due to the increased usage of mobile cellular connectivity. At the same time, the total global electricity usage increases, meaning that the overall energy used by ICT increases and not just its share of the global electricity usage. Different industries are making efforts to be more environmentally friendly and ICT has a significant positive hand print impact in assisting these goals. The ICT sector should do the same for its foot print and try to reduce the resource usage and carbon footprint.

Web pages contain most of their information as text content in an HTML-document. However, most of the data usage comes from images and JavaScript files. The total image and JavaScript bytes per page have multiplied in the past ten years, meanwhile the number of image requests has actually decreased and the number of JavaScript requests has only doubled [3]. This means that the average size of images and JavaScript files has increased significantly. Larger pages require an increasing amount of resources (network bandwidth, storage, CPU) to process, transfer and store on server-side in data centers as well as on end devices, which leads to increased energy consumption and carbon footprint. In addition to image sizes and JavaScripts on web-pages, programmatic online advertising is consuming large amounts of energy while rendering in the end-user devices, but also from the real-time bidding process, which activates hundreds of servers in different data centers during bidding auction [4].

The increased data usage of web services increases the ICT sector's footprint. The main impacts are the increased carbon emissions on the server side and the emissions and energy consumption of mobile networks. The Web Site Carbon [5] calculator estimates the emissions of websites and increased data usage increases the emissions. Mobile networks and the radio base stations show a traffic-related energy consumption profile. This has most recently been measured and presented by Roope Lahti [6]. The measurements show that the base stations' energy consumption grows linearly with the traffic load. As more and more of our daily Internet services are served over mobile networks, a decrease in data consumption has a direct impact on the energy consumption and carbon emissions.

In this paper, we present a study of 10000 most popular web pages according to Google. After removing duplicates, we obtained 7324 unique URLs. We emulated four different user device setups: desktop, scrolled desktop, mobile and scrolled mobile. On each device, approximately 6900 URLs could be connected to. Our results show that 49% to 58% of the average page size was from images, depending on the simulated user device. The second largest type of resource was JavaScript, with 20% to 28% of the average page size.

The largest data savings were related to images. For images, replacing older formats, like PNG and JPEG, with modern formats, such as AVIF and WebP, and resizing images so that their loaded resolution matches their displayed resolution could reduce the amount of data up to 80% or even more. The analysis of JavaScript libraries showed that approximately 5.4% to 5.6% of the total JavaScript was from 16 most popular libraries. Potentially, libraries could be cached in the browser to remove the need for each page to load the same libraries separately in their JavaScript files. Additionally, some data could be saved by removing unused JavaScript.

The Web Sustainability Guidelines (WSG) presents recommendations for making websites and products more sustainable [7]. Sustainability is a wide topic and in terms of environmental aspects, WSG emphasize optimizing web content and infrastructure for energy efficiency, choosing green hosting providers, and enhancing user experience through fast loading times and accessibility. WSG advocates for resource optimization, regular audits, and community engagement to promote sustainable web practices, ensuring a more efficient and environmentally responsible digital ecosystem. Our work can be seen a complementary to the WSG and brings some concrete evidence and data to motivate more careful web page designs and server configurations for a more environmentally sustainable future of ICT.

Next, Section 2 looks at some related work and Section 3 presents the methodology used in this study. Section 4 presents the results regarding the resources of the analyzed pages and Section 5 focuses on the possible data saving methods to reduce page size. Section 6 contains the conclusions and future work.

2 RELATED WORK

HTTP Archive [3] crawls millions of URLs every month and first started gathering this data in 2010. Much of the same data that appears on HTTP Archive, such as total page size, number of requests and the sums of the bytes of different resources on a page, has been gathered in this study. However, HTTP Archive only has page data for non-scrolled desktop and mobile, whereas this study includes the fully scrolled versions as well. Web pages often use lazy loading to load non-blocking, or non-critical, resources only when they are needed. For example, with lazy loading images are only loaded when they come near the viewport. This means that the scrolled version of a web page is often larger than the initially loaded page that has not been scrolled. Furthermore, HTTP Archive does not have any information on how much data is wasted per page and how data could be saved.

Several studies have been made that analyze the characteristics of one country's web, such as Argentina [12] and Spain [13]. Additionally, one study makes a comparison of the characteristics of Chilean and South Korean web [14]. These studies looked at several metrics, such as: the average size of the textual content of pages, pages per website, links between the same and different websites, URL length and URL terms. These studies analyze web pages, but focus more on link structure, the technologies used on the pages and general characteristics rather than the resources and data usage.

Miranda and Gomes [15] compare the results of three earlier studies that analyzed the Portuguese web. Similarly to [12–14], this study primarily focuses on the characteristics as well, but also includes some data on resources. The study shows content size distribution of pages with data from 2003 and 2008. Furthermore, the study compares media type distribution and its trend for all media types and for textual media types only.

Studies by Google [8, 9] show that on mobile, the longer a page takes to load, the more likely a user is going to leave the page. Furthermore, 53% of visits to pages are likely to be abandoned if the page takes more than 3 seconds to load. Reducing page size directly reduces the time it takes to load the page, which makes users less likely to leave the page. Pages that load faster also have a higher conversion rate, for example 0.1 second improvement to page load time on mobile resulted in 8.4% increased conversions and 9.2% higher average order value on retail pages [10]. The focus on the Google work has not been on environmental sustainability.

Some other studies that analyze a large number of web pages focus on privacy aspects [16–19], insecure JavaScript practices [20] or the replicability of web measurement setups [21]. While not directly related to this study, some studies have measured web performance [22] and the quality of experience of web use [23, 24]. The Green Web Foundation also works towards a fossil-free Internet mainly through advocating the use of green energy.

3 MEASUREMENT METHODOLOGY

In our study, we analyzed the top global web pages according to Google's Chrome User Experience Report. Our focus was on the top 10,000 desktop pages based on popularity. Without specifying the form factor (like desktop or mobile), the query returned duplicated URLs with various form factors such as desktop, mobile, and tablet. This yielded 8,769 URLs, but after removing duplicates, we had 7,324 unique URLs. Some pages were inaccessible due to connection or permission issues which lead to 6,887 URLs for desktop, 6,901 scrolled desktop, 6,907 mobile, and 6,895 scrolled mobile pages.

Lighthouse [11] is an open-source tool developed by Google that analyzes web pages and returns data of the page's resources, performance, and practices to improve the page and speed up the page's loading time. Lighthouse was configured to launch an instance of Chromium browser and to return a JSON file for the analyzed page.

For desktop, a resolution of 1920x1080 was used, and for mobile 1080x2400. According to StatCounter [29], 1920x1080 is the most common resolution for desktop devices. The resolution used for mobile was the most common resolution for the best-selling phones for the Finnish operator Elisa [30]. For scrolled desktop and scrolled mobile, the height of the device was set to 30000 pixels (1920x30000, 1080x30000), to simulate scrolling. This makes the page load fully, as if it were scrolled from top to bottom. Some pages can scroll infinitely, so using a high pixel height would set a hard limit on the scrolling and make results comparable. The 30000 pixel height was analyzed to cover all pages expect the infinitely scrolling ones.

Every image that was not in SVG-format (Scalable Vector Graphics) appearing on the pages with the device as desktop was downloaded with Wget [31] to be converted to other formats. SVG-formatted images are XML-based vector images that can not be easily converted to another format. Furthermore, SVG images are already very light-weight and well optimized.

PNG and JPEG are the most widely used image file formats, with 81.1% of web pages using PNG and 76.3% using JPEG [33]. These image file formats are relatively old now and inefficient compared to some modern formats, such as WebP and AVIF that are suggested by Lighthouse. Using a Python script and ImageMagick [27], several versions of these images were created to see the efficiency of compression. An unmodified, original version of each image was kept as a comparison baseline. With Lighthouse, the displayed resolution (that the page shows) and the loaded resolution was acquired for each image. If these resolutions were the same, the image was simply copied, but if the loaded resolution was larger, the image was converted to the displayed resolution. With three quality levels: 60, 80, and 90, WebP-versions of the original images were created. If the original image was already WebP, it was copied to its respective directory. Finally, a version that was converted to WebP, with each quality setting and also resized, was created in its directory.

With Lighthouse, the list of JavaScript libraries used by each page was acquired. Lighthouse utilizes Library Detector for Chrome [32] for this, that detects the JavaScript libraries used on a web page. The sizes of these libraries and their dependencies was obtained with Bundlephobia [28]. The number of dependencies was 0 for 52 libraries, 1 for 18 libraries, 2 for 4 libraries, 3 or 4 for 1 library each and more for the remaining libraries. Generally, most of the libraries should not have overlap with each other based on their dependencies, apart from some larger and less common libraries that had significantly more dependencies. Out of the 126 libraries that Library Detector for Chrome detects, the size of 94 libraries was obtained with Bundlephobia and the remaining 32 libraries returned no value for their size. The total number of occurrences of each library on the pages was counted and the list sorted to get a list showing the libraries from the most common to the least common.

4 PAGE RESOURCE ANALYSIS

Figure 1 shows the average size of various types of resources on analyzed websites. Resources include PNG or JPEG images, GIFs for animations, and JavaScript files that add interactivity and functionality. CSS files style the page, influencing things like fonts and colors. If non-standard fonts are used, they need to be loaded, otherwise, they don't add to the page size. The HTML document contains the page's text, while XHR files handle seamless background data fetching, like loading images or videos. Media files refer to audio and video, and the 'other' category includes miscellaneous data like CORS checks. Third-party files are external resources that can vary widely in type and size. The total size of a webpage is the sum of all these components, except third-party files.

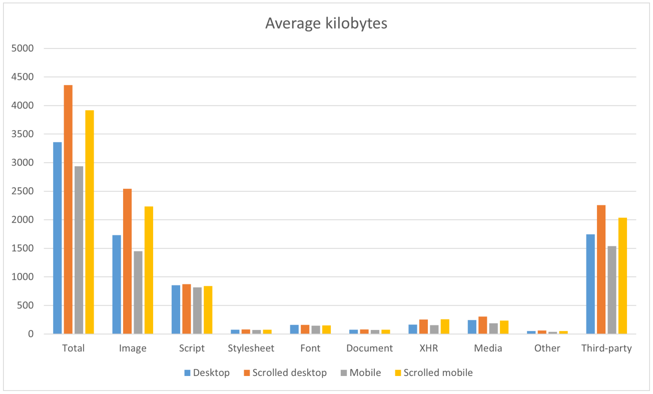

Figure 1: Average size of resources (Kb)

The average page size was 3358 kB on desktop, 4360 kB on scrolled desktop, 2939 kB on mobile and 3918 kB on scrolled mobile. Furthermore, 49% to 58% of the average page size was images. JavaScript on average varied between 20% to 28% of the page size. Combined, images and JavaScript take on average 77% to 78% of the total page size on all devices, leaving the remaining 22% to 23% somewhat equally distributed between style sheets, fonts, document, XHR, media and other files. Approximately 52% of the average page size comes from third-party resources on all devices.

5 DATA SAVING

In this section, data saving methods are discussed. This includes methods suggested by Lighthouse, as well as our analysis of images in more detail and saving possibilities with JavaScript libraries.

5.1 Overview

Lighthouse has recommended several strategies for reducing webpage sizes, as illustrated in Figures 2 and 3. These include removing unused CSS and JavaScript, which are code lines in these files that aren't executed unless the page is interacted with, potentially saving between 376 kB to 407 kB depending on the device. Other methods include eliminating outdated or duplicated JavaScript, applying text compression, and using image compression to lower the quality level from 100 to 85. Additionally, converting older image formats like JPEG and PNG to more efficient ones like WebP or AVIF, resizing images to match their display resolution, and switching large GIFs to video formats can further reduce data usage. Lastly, making data cacheable doesn't reduce the initial load size but decreases the amount of data needed for subsequent visits. These techniques aim to improve loading times and reduce bandwidth usage, enhancing overall user experience and managing data costs effectively.

A significant amount of data could be saved with images. Using compression level of 85 for images, between 179 kB and 285 kB could be saved, without affecting the quality of the images too much [25]. The largest saving opportunity was using better image formats, which would result in 740 kB saved on desktop, 975 kB on scrolled desktop, 634 kB on mobile and 890 kB on scrolled mobile. Lighthouse estimates the data saved by comparing the bytes per pixel of the original image and calculating the expected bytes per pixel if AVIF or WebP formats were used instead of older formats. Another method with large data saving potential is image resizing, where the image's loaded resolution is resized to match the displayed resolution. The average saving potential by resizing images was 594 kB on desktop, 959 kB on scrolled desktop, 235 kB on mobile and 403 kB on scrolled mobile. A small amount of data, 112 kB to 173 kB, could be saved by converting GIFs to video files.

Additionally, on average 764 kB to 1034 kB of data could have a long cache time, but either is not cached at all or has a short cache time. Caching data would reduce the amount of data needed to load when accessing the same page again, which would indirectly reduce the amount of data consumed on web pages.

5.2 Image pixels

Wasted pixels, where the loaded resolution exceeded the displayed resolution, were calculated separately for each image and then added together. If an image was displayed at a higher resolution than it was loaded, it was not counted as wasted because it might appear low-quality.

The results, shown in Table 1, reveal that the mean wasted-to-loaded pixel ratio exceeds 90% across all devices when calculated using the mean. The median ratio is around 30% on desktops and over 50% on mobile devices. This discrepancy suggests that images are often oversized for their display context, particularly on mobile, where images are loaded at desktop sizes but displayed at much smaller mobile resolutions. We also found that scrolling devices tend to load and display more pixels, but the mean number of wasted pixels remains similar across scrolled and unscrolled devices.

The cumulative distribution function (CDF) of the wasted-to-loaded pixel ratio across different device types is illustrated in Figure 4. For desktop and scrolled desktop configurations, the CDF exhibits a nearly linear progression, indicating a more uniform distribution of wasted pixels relative to pixels loaded. In contrast, the mobile and scrolled mobile configurations demonstrate a CDF that closely resembles an exponential function, suggesting a significant variation in the efficiency of pixel usage across pages. This combined visualization highlights the distinct patterns of pixel wastage across device types, emphasizing the need for optimized content delivery tailored to each device's characteristics.

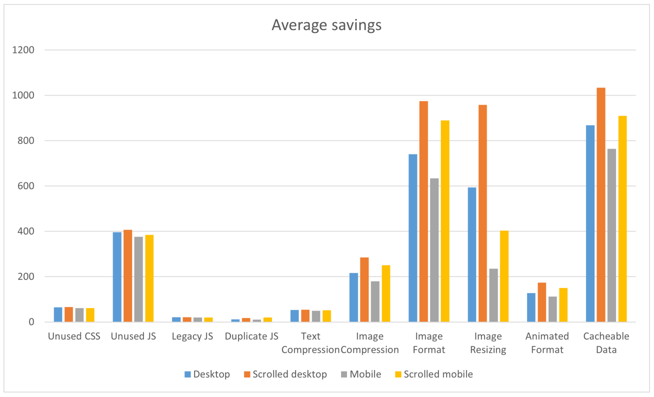

Figure 2: Average savings (Kb) reported by Lighthouse

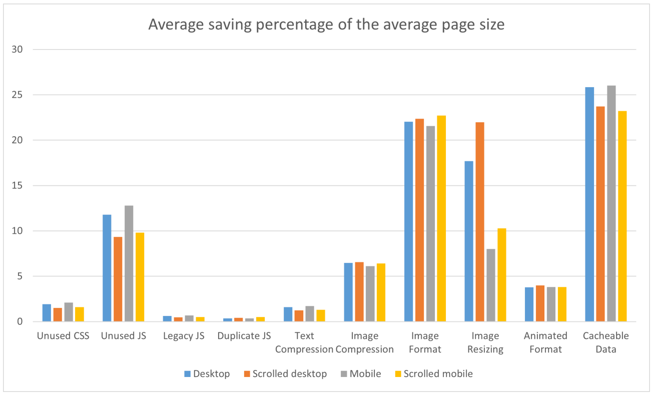

Figure 3: Average saving percentage of the average page size

Table 1: Mean and median pixels for each device

	Desktop	Scrolled desktop	Mobile	Scrolled mobile
Mean pixels loaded	40,485,326	41,998,330	36,720,507	38,330,907
Median pixels loaded	2,541,122	3,895,472	1,767,237	3,250,553
Mean pixels displayed	3,376,376	13,559,285	1,580,313	6,736,936
Median pixels displayed	1,597,252	2,604,223	560,093	1,145,325
Mean wasted pixels	37,983,881	38,126,610	35,391,672	36,236,343
Median wasted pixels	767,026	1,108,309	1,028,193	1,722,080
Mean wasted-to-loaded ratio	93.8%	90.8%	96.4%	94.5%
Median wasted-to-loaded ratio	30.2%	28.5%	58.2%	53.0%

This shows the same trend as the median wasted-to-loaded ratio in Table 1, which is that the ratio is higher on mobile and scrolled mobile than on desktop and scrolled desktop.

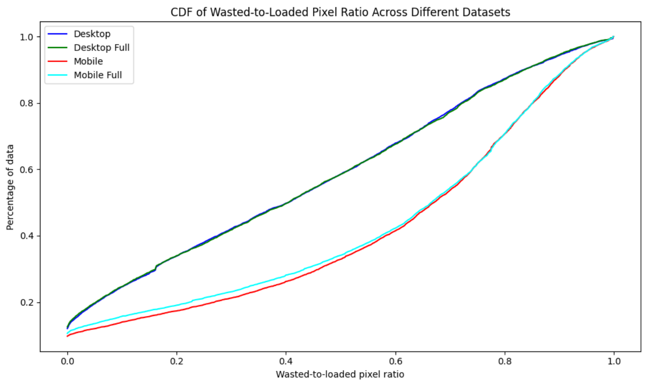

Figure 4: Distribution of Pixel Ratio

5.3 Image data saving in more detail

Table 2 shows the size of all images downloaded from the pages with the device as desktop with different formats. Original is the size of the images as they appear on the pages.

Web pages can load an image with a certain resolution, for example 100x100, but display this image in a different resolution, for example 50x50. If the loaded resolution is larger, some transferred data is wasted in the rendering. Data can be saved by resizing images to match the displayed resolution. In the table, this appears in the 'Resized' row. Over half of the total size could be saved by resizing, which shows that a large number of images are displayed with a smaller resolution than they are loaded with on web pages.

WebP is a modern image format that offers better compression than PNG or JPEG [34], which results in smaller file sizes for images. According to Google's studies, WebP lossless images are 26% smaller in file size compared to PNGs [35] and WebP lossy images are 25-34% smaller in file size than JPEG images [36]. Another study [37] found that WebP has a good compression efficiency when compared to JPEG codecs and has a lower peak signal-to-noise ratio, with natural images with lower bitrate. But with higher bitrate and for synthetic images, such as digital patterns, WebP's efficiency decreases. Netflix did a study [38] that compared different image formats, including different versions of JPEG that maximize a specific metric, WebP and AVIF minimizing mean squared error and AVIF maximizing Structural Similarity Index (SSIM). SSIM measures the similarity between images. The study used several metrics, such as SSIM, Multi-Scale Structural Similarity (MS-SSIM), Visual Information Fidelity (VIF) and peak signal-to-noise ratio (PSNR) to compare the bitrate difference to the original PNG image. The study found that for all metrics, WebP performed better, meaning it had a lower bitrate when compared to the different JPEG versions or the original PNG. Furthermore, AVIF performed even better than WebP with all metrics.

The best image format could be chosen by web server upon receiving the information about the user agent and capabilities. Thus, newer web browsers receive the most data-efficient images and older browsers then left with the less optimized image formats.

In our study, only the conversion to WebP was considered, despite AVIF being more efficient, because WebP is more common: WebP is used on 5.1 % of all websites and AVIF in less than 0.1% according to W3Techs' data [33]. Furthermore, WebP is more widely supported by browsers than AVIF. In Table 2 WebP formatted images are 'WebP 60', 'WebP 80' and 'WebP 90', where the number references the quality level. A quality level higher than 90 would often lead to the file size increasing, because of lossy images being converted to lossless or lossy images being converted to lossy with higher quality than the original image. Quality levels of 80 and 90 produce images that are visually nearly indistinguishable from the original, while saving a significant amount of data. Quality level 60 saves even more data, but looks noticeably different, especially with large images.

By combining these methods, resizing and using WebP format, even more data can be saved. Even with quality level 90, resizing and using WebP, lowers the total data used to less than a third of the original.

The number of images in each category in Table 2 differs slightly. The reason is that some of the original images could not be downloaded properly, which resulted in an empty image file of 0 bytes. Furthermore, for some images, the conversion to WebP did not succeed.

Table 2: Total Downloaded Image Sizes by Format: Desktop.

Format	Total size	Number of images
Original	12.1 GB	245483
Resized	5.9 GB	234810
WebP 60	4.3 GB	227668
WebP 80	5.3 GB	227668
WebP 90	6.7 GB	227668
Resized WebP 60	2.2 GB	224930
Resized WebP 80	2.7 GB	224930
Resized WebP 90	3.5 GB	224930

5.4 JavaScript libraries

A theoretical method to reduce the amount of data loaded when a page is loaded, is to cache some of the common JavaScript libraries to the browser, to prevent the same libraries getting downloaded multiple times as many pages use the same common libraries. This would cause issues if some of the libraries were not backwards compatible with their older version, in which case multiple versions of the same library would have to be cached.

The most common libraries were jquery (used on 4349 pages), corejs (3701), react (1295), boostrap (1161) and jquery_ui (1035). Thus, over half of the pages use jquery and slightly fewer, but still over half of the pages use corejs.

The proportion of JavaScript attributed to libraries on initially loaded pages is depicted for both desktop and mobile devices in Figure 5, showing data for pages loaded once ("Desktop" and "Mobile") alongside pages that are fully scrolled ("Desktop Full" and "Mobile Full"). All configurations exhibit similar trends across different datasets. Notably, the 16 most used libraries account for JS savings of approximately 6% to 9%, varying slightly between the different device configurations. Beyond the top 16 libraries, the savings percentage levels off, remaining consistent up to around 9%. Implementing caching for these libraries could represent significant data savings. However, these percentages may not fully reflect typical user behavior, as users often revisit fewer pages repeatedly rather than accessing many different websites. Thus, the practical savings could potentially be greater than those illustrated here.

6 CONCLUSION AND FUTURE WORK

In our study, we observed that the majority of the page size, ranging from 49% to 58%, was attributed to images, and about 22% to 23% to JavaScript, amounting to a combined 77% to 78% of the total, contingent on the device used. Notably, a significant proportion of page pixels were transmitted but not eventually used, especially on mobile and scrolled mobile devices. This indicates that many images are loaded at resolutions higher than their display specifications. The predominant data-saving recommendations from Google Lighthouse were image-centric. By adopting modern image formats and resizing images to align loaded resolution with display resolution, substantial savings can be realized. As an illustration, initially loading every image totaled 12.1 GB of data. After resizing and transitioning to WebP at a 90% quality level, the data dramatically dropped to 3.5 GB. Potential savings were also identified in omitting unused JavaScript and caching JavaScript libraries, reducing the

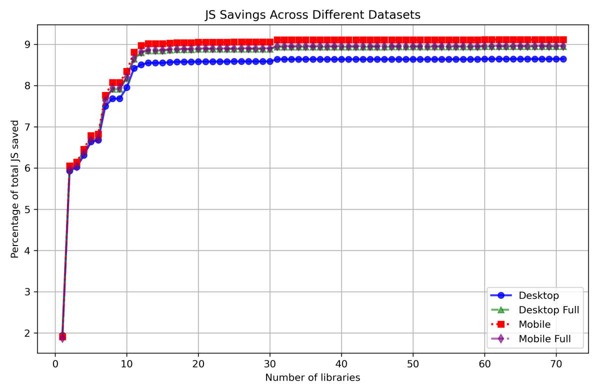

Figure 5: JavaScript Library Percentages: Across all datasets

need for repetitive loads across diverse pages. In some cases, tailored solutions could negate the need for specific libraries, further optimizing data use.

The main environmental benefits from smaller pages come from reduced emissions on the server side and mobile networks. Mobile networks are energy-proportional to a great extent and thus less data will save energy and create less emissions. Also, if load on mobile networks could drop or even grows slower, there is less pressure on increase capacity by installing new hardware, that needs to be built and powered.

We see many directions for future work. One clear topic is the overhead of the HTTP/HTTPS signalling and data transfer. From the early days of the World Wide Web when web sites were rather simple and content was mostly centralized, current web sites are built using over 100 different objects coming from tens of different servers. Each connection requires its own HTTP/HTTPS handshake and the overhead of the all the signalling would be interesting to measure and understand. A related questions is the benefit of HTTP pipelining with the web services built today.

In our analysis, we found numerous cases where a web site had extremely heavy content, such as 30MB pictures sent to the device that were downgraded on the browser to small illustrations, or 100MB animated GIF files that were used instead of videos, probably to enable more universal playback. Our data collection could be much more fine-graded, looking at daily changes in web site content, and what extreme cases we find. For example, studying the structure of a given set of web sites on an hourly basis and analyze the variance and reasons for that.

Developers and publishers use various tools to build the web sites. Can we identify the tools used and is there a pattern that some tools create heavier and more complex web sites, while other tools produce lighter and more sustainable end results? Could these designer tools somehow support more sustainable development of web-based services? The ultimate goal of our works is to raise awareness in the ICT sector on more careful design of services to enable a more sustainable future that does not require a constant and steady increase in capacity and eventually growth in the consumption of energy and natural resources. We see our work complementary to the Web Sustainability Guidelines (WSG) and brings evidence and support for more environmental web services.

The analysis presented in this paper resulted in an open tool that performs analyses and recommendations to web site developers. Our goal is to provide the data acquired through the service openly to researchers and the public at large. The public service can be found from greenpages.aalto.fi.

REFERENCES

[1] Energy consumption of ICT. https://post.parliament.uk/research-briefings/post-pn-0677/

[2] Andrae, A., & Edler, T. (2015). On Global Electricity Usage of Communication Technology: Trends to 2030. *Challenges*, 6(1), 117–157. DOI 10.3390/challe6010117

[3] HTTP Archive. (2022, October 22). *All Reports.* https://httparchive.org/reports

[4] Pärssinen, m., Kotila, M., Cuevas, R., Phansalkar, A., & Manner, J. (2018). Environmental impact assessment of online advertising. *ENVIRONMENTAL IMPACT ASSESSMENT REVIEW*, 7(1), 177–200. DOI 10.1016/j.eiar.2018.08.004

[5] https://www.websitecarbon.com/

[6] Roope Lahti (2023). Assessing 5G and multi-access edge computing energy efficiency for industrial applications. Aalto University, Finland.

[7] https://www.w3.org/blog/2023/introducing-web-sustainability-guidelines/

[8] Google. (2016, September 8). The need for mobile speed. https://www.blog.google/products/admanager/the-need-for-mobile-speed/

[9] Google. (2018, February). Find out how you stack up to new industry benchmarks for mobile page speed. https://www.thinkwithgoogle.com/marketing-strategies/app-and-mobile/mobile-page-speed-new-industry-benchmarks/

[10] Deloitte. *Milliseconds Make Millions: A study on how improvements in mobile site speed positively affect a brands bottom line.* https://www2.deloitte.com/ie/en/pages/consulting/articles/milliseconds-make-millions.html

[11] Google. (2022, September 20). *Lighthouse* [Source code]. https://github.com/GoogleChrome/lighthouse

[12] Tolosa, G., Bordignon, F., Baeza-Yates, R. & Castillo, C. (2007). Characterization of the Argentinian Web. *Cybermetrics: International Journal of Scientometrics, Informetrics and Bibliometrics*, 11(1), ISSN 1137-5019

[13] Baeza-Yates, R., Castillo, C. & López, V. (2005). Characteristics of the Web of Spain. *Cybermetrics: International Journal of Scientometrics, Informetrics and Bibliometrics*, (9), ISSN 1137-5019

[14] Baeza-Yates, R., Castillo, C., Lalanne, F. & Dupret, G. (2004). Comparing the Characteristics of the Korean and the Chilean Web. Retrieved from https://chato.cl/papers/baeza_04_comparing_chilean_web_korean_web.pdf

[15] Miranda, J. & Gomes, D. (2009). Trends in Web Characteristics. *2009 Latin American Web Congress*, 146-153. DOI 10.1109/LA-WEB.2009.28

[16] Englehardt, S. & Narayanan, A. (2016) 'Online Tracking: A 1-million-site Measurement and Analysis', *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16)*, Association for Computing Machinery, New York, NY, USA, pp. 1388–1401. DOI 10.1145/2976749.2978313

[17] Acar, G., Eubank, C., Englehardt, S., Juarez, M., Narayanan, A., & Diaz, C. (2014). The Web Never Forgets: Persistent Tracking Mechanisms in the Wild. *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (CCS '14)*. Association for Computing Machinery, New York, NY, USA, 674–689. DOI 10.1145/2660267.2660347

[18] Papadogiannakis, E., Papadopoulos, P., Kourtellis, N., & Markatos, E. P. (2021). User Tracking in the Post-cookie Era: How Websites Bypass GDPR Consent to Track Users. *Proceedings of the Web Conference 2021 (WWW '21)*. Association for Computing Machinery, New York, NY, USA, 2130–2141. DOI 10.1145/3442381.3450056

[19] Pujol, E., Hohlfeld, O., & Feldmann, A. (2015). Annoyed Users: Ads and Ad-Block Usage in the Wild. *Proceedings of the 2015 Internet Measurement Conference (IMC '15)*. Association for Computing Machinery, New York, NY, USA, 93–106. DOI 10.1145/2815675.2815705

[20] Yue, C. & Wang, H. (2013). A measurement study of insecure javascript practices on the web. *ACM Transactions on the Web*, 7(2), 1-39. DOI 10.1145/2460383.2460386

[21] Demir, N., Große-Kampmann, M., Urban, T., Wressnegger, C., Holz, T., & Pohlmann, N. (2022). Reproducibility and Replicability of Web Measurement Studies. *Proceedings of the ACM Web Conference 2022 (WWW '22)*, 533–544. DOI 10.1145/3485447.3512214

[22] de Saxcé, H., Oprescu, I. & Chen, Y. (2015) Is HTTP/2 really faster than HTTP/1.1? *2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 293-299, DOI 10.1109/INFCOMW.2015.7179400.

[23] Asrese, A.S., Walelgne, E.A., Bajpai, V., Lutu, A., Alay, Ö., & Ott, J. (2019). Measuring Web Quality of Experience in Cellular Networks. *Passive and Active Measurement: 20th International Conference, PAM 2019*, 11419, Puerto Varas, Chile. DOI 10.1007/978-3-030-15986-3_2

[24] Bocchi, E., De Cicco, L., & Rossi, D. (2016). Measuring the Quality of Experience of Web users. *SIGCOMM Comput. Commun. Rev.* 46(4), 8–13. DOI 10.1145/3027947.3027949

[25] https://developers.google.com/speed/webp/docs/webp_study

[26] Google. (2022, September 20). *BigQuery.* https://cloud.google.com/bigquery

[27] ImageMagick Studio LLC. (17.9.2022). *ImageMagick.* https://imagemagick.org/script/index.php

[28] Kanodia, S. (10.10.2022). *Bundlephobia* [Source code]. https://github.com/pastelsky/bundlephobia

[29] StatCounter. (2022, November 25). *Desktop Screen Resolution Stats Worldwide.* https://gs.statcounter.com/screen-resolution-stats/desktop/worldwide

[30] Elisa. (2021, May 3). *Elisan myydyimmät puhelimet ja älykellot huhtikuussa 2021.* https://elisa.fi/yhtiotieto/uutishuone/tiedotteet/elisan-myydyimm%C3%A4t-puhelimet-ja-%C3%A4lykellot-huhtikuussa-2021/67147864089084/

[31] Free Software Foundation, Inc. (2022, September 17). *GNU Wget.* https://www.gnu.org/software/wget/

[32] Bredow, A., & Michel J. (2022, October 10). *Library Detector For Chrome* [Source code]. https://github.com/johnmichel/Library-Detector-for-Chrome

[33] W3Techs. (2022, October 11). *Usage statistics of image file formats for websites.* https://w3techs.com/technologies/overview/image_format

[34] Google. (2022, September 18). *An image format for the Web.* https://developers.google.com/speed/webp

[35] Google. (2022, October 8). *Lossless and Transparency Encoding in WebP.* https://developers.google.com/speed/webp/docs/webp_lossless_alpha_study#results

[36] Google. (2022, October 8). *WebP Compression Study.* https://developers.google.com/speed/webp/docs/webp_study

[37] Ginesu, G., Pintus, M., & Giusto, D. D. (2012). Objective assessment of the WebP image coding algorithm. *Signal Processing: Image Communication*, 27(8), 867-874. DOI 10.1016/j.image.2012.01.011

[38] Mavlankar, A., De Cock, J., Concolato, C., Swanson, K., Moorthy A., & Aaron, A. (2020, February 14). AVIF for Next-Generation Image Coding *Netflix Technology Blog.* https://netflixtechblog.com/avif-for-next-generation-image-coding-b1d75675fe4

Received 29 September 2023; revised 12 March 2009; accepted 5 June 2009