

Byte-By-Night: Web Cache Object Forwarding from Desktop to Mobile for Energy Consumption Optimizations

Troy Johnson
Department of Computer Science
Central Michigan University
Mount Pleasant, MI 48859
johns4ta@cmich.edu

Patrick Seeling
Department of Computer Science
Central Michigan University
Mount Pleasant, MI 48859
pseeling@ieee.org

Abstract—

*Index Terms—*Mobile communication; Middleware; Energy consumption

I. INTRODUCTION

With the beginning of the 21st century, new support for mobile connected user devices has fueled a continuous increase in the demand for mobile data. Web requests now originate in a majority from wirelessly connected user devices, a trend that is likely to continue in the foreseeable future [?]. Simultaneously, the overall user behavior and demand for more rich media inclusion into web pages has increased the overall amount of data that is required to be transmitted per page, see, e.g., [?]. Caching on the client side has been effectively used in the past and was, together with increased numbers of parallel object downloads, able to decrease wait times for desktop clients as reported in [?].

The remainder of this paper is structured as follows. In the following section, we describe our overall scenario before we provide an example based on a particular web page. In the subsequent Section IV, we discuss the overall properties of a large web page dataset archive and provide a simulation-driven evaluation of our approach. We conclude in Section ??.

II. SCENARIO DESCRIPTION

III. INDIVIDUAL EXAMPLE

In this section, we outline an individual example for the German news web page for “Der Spiegel,” accessible at <http://www.spiegel.de> as an example for a rich media and advertising material containing web presence that is frequently updated. On [insert final date], we performed a speed test online using Internet Explorer as browser instance for a desktop client, Chrome as mobile browser on a Google Nexus 5 instance, and Safari for an iPhone 4 instance provided by WebpageSpeedTest.org (both mobile clients were traffic shaped to 3G connection emulations). An example screenshot of the web page as rendered is provided in Figure 2.

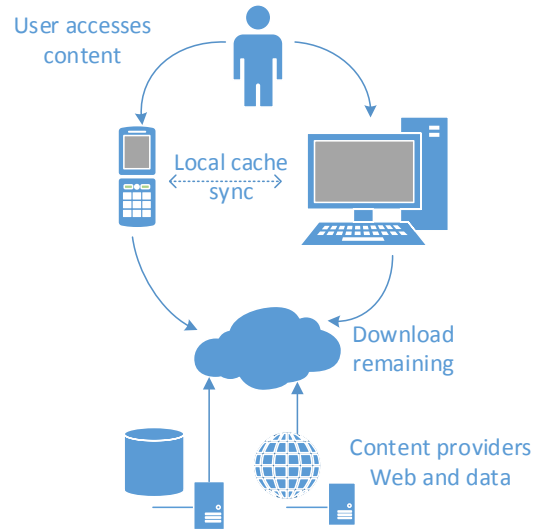


Fig. 1. Overall setup of world wide web based content access by a user over time using primary and secondary devices to display the same content. If the devices are able to synchronize their caches through short-range communications, energy savings become possible.

A. Metrics

We evaluate the reported data as follows. For each display modality, we gather the individual web objects requested and the response sizes and headers for caching information (for HTTP response codes 200 only, as we are not interested in redirects). We denote the number of all objects returned for a request modality (i.e., Internet Explorer for the desktop client example and iPhone or Nexus 5 for mobile counterparts) as N , their average size as \bar{X} , the standard deviation of their sizes as σ_X , and the Coefficient of Variation (CoV) of the returned object sizes as CoV_X . As in the HTTP specification, the *max-age* directive overrides others, so we initially consider that directive for the cache longevity of the individual objects. If no explicit information is found, we consider the header’s *expires* information, which provides a secondary cache lifetime. If



Fig. 2. Screenshot of rendered webpage from WebPageSpeedTest.org. As illustrated, the main textual and pictorial content items are flanked by background and interactive advertisements.

TABLE I
HIGH-LEVEL OVERVIEW OF DIFFERENT CLIENT WEBPAGE STATISTICS FOR
HTTP://WWW.SPIEGEL.DE

Statistic	IE Desk.	iPhone	Nexus 5
N	154	132	145
\bar{X}	10402.77	8846.39	10201.63
σ	19103.19	13438.05	25684.58
CoV_X	1.84	1.52	2.52
\bar{T}	106285.14	114780.69	114818.28
σ_T	125080.27	125576.65	126090.90
CoV_T	1.18	1.09	1.10

both are not found, or if the *expires* date is in the past or right at the request time, we set the cache lifetime we consider here to zero. Similar to the notation for the object sizes, we denote the expiration time characteristics as \bar{T} , σ_T , and CoV_T , respectively. In case of size differences between the three, we utilize the smallest size as lower boundary.

B. Data Description

We provide an initial high-level overview of the webpage characteristics for <http://www.spiegel.de> as requested in Table I.

We initially observe that the desktop version (with Internet Explorer 9 as requesting browser client) exhibits the highest number of objects, followed by the mobile Chrome/Android and Safari mobile/iOS versions, respectively. Interestingly, there is only a minor difference in the number of objects between these versions. Next, we observe that the trend for the average number of bytes is aligned with the number of elements. In total, the mobile Chrome version is almost on par with the desktop one (at 92%) and only the Safari mobile version seeming optimized (at 72%). The standard deviation and Coefficient of Variation (CoV) amongst individual element sizes are highest for the mobile Chrome version, indicating more significant size differences than for the safari version, which is one unit lower, while the desktop version falls into the middle.

Next, we evaluate the elements' caching properties on a high level in Table I. Overall, we note that the highest average caching time is observed for the mobile Chrome access with the Nexus 5 device, trailed immediately by the iOS access and with distance by the desktop browser. We note that the overall variability in terms of expiration times amongst elements is fairly low and comparable, as indicated by CoV values around 1.1.

C. Evaluation of Local Cache Synchronization

We now shift the view to the possibility for identical objects to be re-utilized locally to avoid additional download penalties in time and energy consumption. Simultaneous with a web page request, a local broadcast could “ask” for the cached elements for a website to be delivered locally from participating clients. Alternatively, cellular provisioning methods, such as in LTE-A, could initiate the device-to-device local exchange as well. Once the request is received, the local coordination can take place, which inherently results in the forwarding of elements that are the same across browser instances and which have a future cache lifetime expiration.

We filter out elements that are not similar amongst the different requesting devices, which results in 82 objects with the same URL and size combination. While we note an identical cache lifetime for most of these, a slight increase is notable from the Desktop over Safari to Chrome. Next, we remove items that exhibit a cache lifetime of zero for any of the three requesting clients, resulting in 59 objects, which now exhibit a reduced average cache lifetime for the mobile browsers, whereby mobile Chrome exhibits the shortest. In turn, we choose the iOS Safari mobile as our exemplary base for calculations. We utilize the approximations for energy consumptions presented in [1] to determine the amount of energy required as time progresses if local data exchange can be performed using either Bluetooth or WLAN technologies. As illustrated, the complete download energy for cellular data is the upper limit on energy spent downloading the data associated with the web page. The Bluetooth and WLAN data exchanges with other local clients both follow the same underlying trend with increasing energy required to transmit all data as time progresses from the last access point in time.

We illustrate the relative amount of data/items within the cache in addition to the attainable savings in Figure 4 as a function of delayed access time. We observe that a significant amount of the overall data can be cached (and would in turn be suitable for localized sharing). We additionally observe that even for longer durations up to multiple days, significant amounts of data remain in the cache. Combining the streaming from local clients through either Bluetooth or WLAN technologies, we utilize the approximations outlined in [] to derive the relative gains of our approach in terms of energy consumption. Specifically, we compare our approach (combining locally exchanged caches and cellular downloads of the remainder) with the regular download of all items through the cellular network interface.

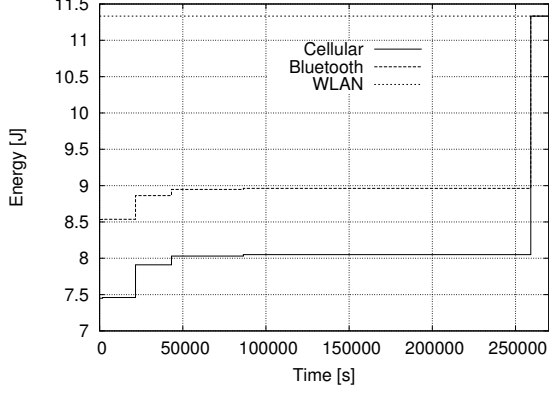


Fig. 3. Energy required to download web page data via cellular connection or through combination with partial local exchange with Bluetooth/WLAN.

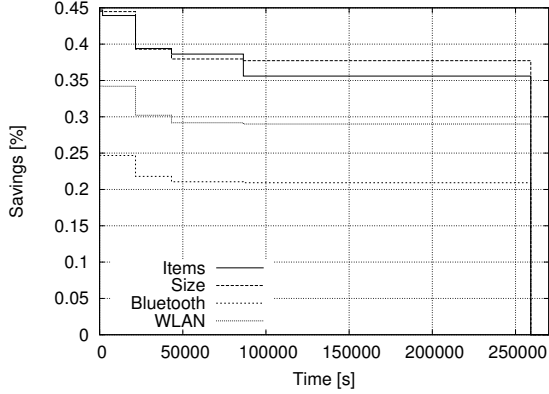


Fig. 4. Relative number of items and data amounts in cache and savings resulting from local data exchange with

IV. LARGE DATASET PERFORMANCE EVALUATION

We now evaluate our proposed approach through simulations based on a large dataset of landing web pages of the most popular websites. For this purpose, we retrieved the publicly available web page response details from *httparchive.org* and create a subsequent dataset for the October 1st, 2013 dataset.

A. Dataset Description

The dataset we utilize for the performance evaluation contains the individual web responses for the most popular websites of fixed (Internet Explorer for the desktop) and mobile (iOS for iPhone 4) websites, ranked through the Alexa popularity index. Furthermore, the dataset contains the individual response details, including cache expiration times and sizes.

Similar to the individual web page example, we provide an overall description of the response sizes and maximum

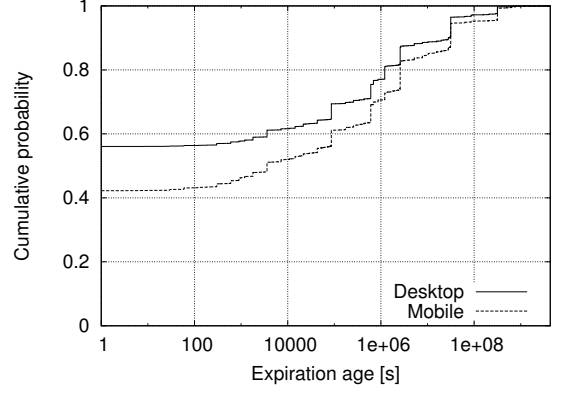


Fig. 5. Cumulative probability

expiration ages in Table II. We note that the overall average response size as result of desktop requests (when batched into the outlined time frames) is around of 17 kB, with the largest average sizes occurring between one and twelve hours. Mostly, we note that the majority of objects exhibit a short expiration time between zero and 30 seconds. For the responses to mobile devices, we note that the majority of objects are in the same short time span and the long time span of more than one day. The highest average response sizes, however, are observed between one hour and half a day, in difference to the desktop counterpart.

We illustrate the cumulative distribution for the desktop and mobile client response expiration ages in Figure 5. We observe that a significant portion ($> 40\%$) of responses from web servers exhibits an immediate expiration and would, in turn, require an immediate retrieval by clients. The cumulative expiration probability is higher for responses to desktop clients, which can be attributed to the typically more abundant bandwidth (and less optimization requirements) in contrast to a mobile setting (where web content developers might be more considerate of limitations).

B. Simulation Model Description

We evaluate the effectiveness of our approach through simulation, which we perform as follows. For the time period of one week, we randomly select a web page from the mobile dataset and separate the data that is required to be downloaded and the data that is still available in cache due to the maximum expiration age of the individual objects. We subsequently wait for a random amount of time before repeating the procedure. We now describe the process in greater detail.

Initially, we sort all landing web pages $l, l = 0, \dots, L$ from the Oct. 1 dataset in the *httparchive.org* dataset based on their web page popularity ranking. To simulate the popularity of web pages, it was found that the Zipf distribution effectively describes the popularity of individual requests, see, e.g., [1].

TABLE II

Category	Desktop (IE)				Mobile (iOS)			
	$\sum X_i$ [MB]	N	$\sigma(X_i)$ [kB]	\bar{X}_i [kB]	$\sum X_i$ [MB]	N	$\sigma(X_i)$ [kB]	\bar{X}_i [kB]
$\text{expAge} < 30 \text{ Seconds}$	238490.28	15670679	1698.59	15.22	1355.24	122598	582.98	11.05
$30 \leq \text{expAge} < 1 \text{ Minute}$	170.94	29130	27.38	5.87	12.68	898	70.38	14.12
$1 \text{ Minute} \leq \text{expAge} < 1 \text{ Hour}$	15192.40	797450	270.06	19.05	282.36	15854	226.62	17.81
$1 \text{ Hour} \leq \text{expAge} < 12 \text{ hours}$	29798.68	1186947	751.45	25.11	244.14	17424	209.84	14.01
$12 \text{ hours} \leq \text{expAge} < 1 \text{ day}$	6188.57	358998	227.63	17.24	91.37	5952	91.37	15.35
$1 \text{ day} \leq \text{expAge}$	195784.64	9888147	1337.36	19.80	2122.96	126629	750.51	16.77
Average	80937.59	4655225	718.75	17.39	684.79	48226	321.95	14.20

We chose to utilize the non-modified Zipf distribution for our simulation purposes, noting that additional modifications to the distribution exist that can further enhance it, see, e.g., [2]. For the total number of web pages L , we derive the probability of page l being selected in turn as

$$p(l) = \frac{l^{-\alpha}}{\zeta(l)}, \quad (1)$$

whereby $\zeta(\cdot)$ denotes the Zeta function. We furthermore set $\alpha = 0.85$ as an average choice in common ranges utilized for this parameter.

Once the individual mobile web page l was randomly chosen, we compare it to its desktop counterpart, with $\{d, m\}$ denoting the request modality. As each page l exhibits a time-sensitive number of responses, we denote them as $r^{\{d, m\}}(l, t)$ and each page exhibits $R^{\{d, m\}}(l, t = 0)$ in total. Let $t_c(r^{\{d, m\}}(l))$ denote the expiration or max-age directive received with the response at $r^{\{d, m\}}(l, t = 0)$. The number of objects or responses to be retrieved at time t in turn are given as

$$R^{\{d, m\}}(l, t) = \sum_{r^{\{d, m\}}(l, 0)} [t_c(r^{\{d, m\}}(l)) \geq t] \quad (2)$$

where $[\cdot]$ denotes the Iverson Bracket. In the following, we abbreviate to t_c for readability when in direct context.

We furthermore denote the size of the individual response retrieved at time t as

$$x_r^{\{d, m\}}(l, t) = x_r^{\{d, m\}}(l, 0) \cdot [t_c \geq t]. \quad (3)$$

The total size of objects or responses retrieved at simulation time t thus is given as $\sum_{r^{\{d, m\}}(l, 0)} x_r^{\{d, m\}}(l, t)$.

For performance evaluation purposes, we simulate the user behavior similar to the process outlined in [3], whereby we randomly draw the time between requests t_u as Pareto distributed using $p(t_u) = \beta k^\beta t_u^{-(\beta+1)}$ with $\beta = 1.5$, $k = 30$. We simplifyingly assume that no delays are accrued due to instantaneous cache retrievals or downloads. Using time index i , we thus derive $t_{i+1} = t_i + t_u$. We continue the simulation until $t_{i+1} = t \geq T = 640800$ [s], i.e., one week.

C. Simulation Results

Cache Simulation (.py) Spreadsheet contains data of mobile and fixed pages that shared identical urls -contains columns

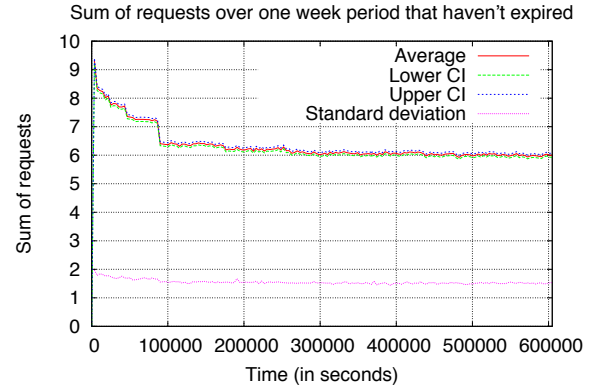


Fig. 6. Sum of requests over time

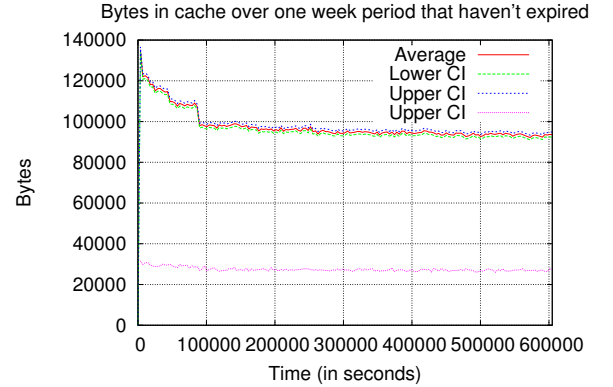


Fig. 7. Sum of requests over time

that shows the properties of identical requests that exist between fixed and mobile pages and properties of that request -spreadsheet sorted based on mobile page rank

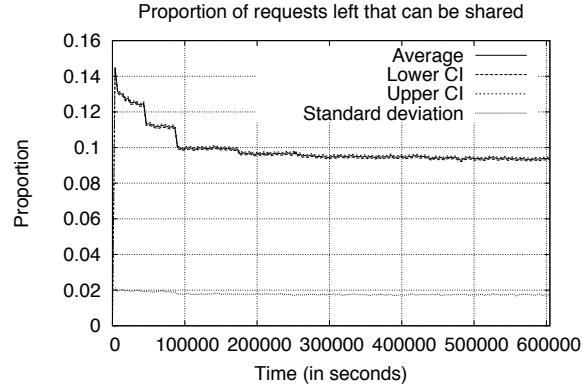


Fig. 8. multiplied

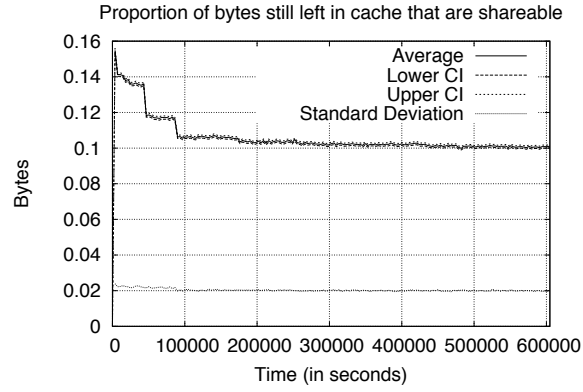


Fig. 9. multiplied

V. CONCLUSION

REFERENCES

- [1] G. P. Perrucci, F. H. P. Fitzek, and J. Widmer, "Survey on energy consumption entities on the smartphone platform," in *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd*, May 2011, pp. 1–6.
- [2] S. A. Krashakov, A. B. Teslyuk, and L. N. Shchur, "On the universality of rank distributions of website popularity," *Computer Networks*, vol. 50, no. 11, pp. 1769 – 1780, 2006. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1389128605002513>
- [3] G. Anastasi, M. Conti, E. Gregori, and A. Passarella, "Performance comparison of power-saving strategies for mobile web access," *Performance Evaluation*, vol. 53, no. 3, pp. 273–294, 2003.