Greening the Internet surf:
Experimental measurements of Web power-consumption

Aruna Prem Bianzino, Anand Kishore Raju, Dario Rossi

Telecom ParisTech, France
firstname.lastname@enst.fr

***Abstract*—Nowadays, Web and cloud services have become irreplaceable cornerstones of both enterprise productivity and user entertainment. While the fact that Web permeates many aspects of our lives is no secret to anyone, the power-consumption induced by these services is not obvious either. In this work, we gauge the power-consumption of end-user PCs browsing the Web, with special attention to Flash technology, and offer a very simple yet effective solution to limit unnecessary power expenditures.**

I. Introduction and motivation

Since the mass-market adoption of networking technology, the Web has often been used as a synonym for the Internet. In recent years, renewed user interest for the Web comes from cloud services such as GoogleDocs, multimedia-rich portals as YouTube or Megavideo, and social networking as Facebook or Twitter, just to name a few representative examples.

While we still refer to this rich set of applications simply as the Web, clearly much has changed in terms of the underlying technology: considering client-side only, Adobe Flash is one of the latest examples of this evolution, and perhaps the most important according to its penetration rate [1]. Often, Flash has been criticized for being CPU hungry, which led to a number of comparative tests against alternatives such as Silverlight [2] and, more recently, HTML5 [3]. Due to its popularity, we focus on Flash as an example of recent Web technology: at the same time, the conclusion we gather in this work are of more general extent, as they are rooted in current Web browsing practice.

A few studies [4], [5] already exist that measure the average power-consumption of Websites (focusing on dynamic content [4] or battery lifetime [5]) hence providing enough anecdotal evidence of the existing correlation between CPU load and power usage. Along similar lines, in this work we evaluate the power-consumption of the Web from the end-user point of view, by means of a thorough measurement campaign that takes into account the variability of Websites, browsers, operating system and hardware equipment.

II. Methodology

At high-level, the power drain P of any end-user device can be divided in a fixed component P_0, that represents the power-consumption in idle state, and a variable component proportional to the sustained workload α, i.e., $P = P_0 + f(\alpha)$. In order to thoroughly assess the power-consumption with an experimental methodology, several factors need to be taken into account, such as: hardware (HW), operating system (OS), browser settings (SW) and benchmark Websites (Web). The fixed component P_0 can be affected by both HW (e.g., CPU frequency, voltage scaling, video-card GPU, etc.) and OS (e.g., tickless kernel). Web content and SW instead affect the variable component (with scripts contained in Webpages generating a different workload level depending on the complexity of the tasks to be performed), and may interact with other factors (e.g., SW optimized for specific OS).

In this work, we consider 3 PCs (1 desktop and 2 laptops), 3 operating systems (Windows, OS X and Linux), 4 browsers (the latest stable versions of Internet Explorer, Firefox, Chrome and Safari), 2 browser settings (Flash enabled/disabled), and 14 websites (corresponding to different content categories). We repeat each experiment twice, for a total of more than 500 experiments, by browsing to a specific Webpage and measuring the power drain of our end-system during a fixed time-frame. We instead neglect the energy consumed by data-centers and by the Internet data transfer, for which we refer the interested reader to [6] and [7] respectively.

Notice that our measurement campaign balances the variability in the *Platform* (i.e., the HS, OS and SW settings) to that in the benchmark *Websites*: as shown in Fig. 1 and Tab. I, we consider N_P=18 platform configurations and explore N_W=14 websites. Notice that, while we do not explore the full cross product of settings (e.g., we avoid weird configuration such as Safari over Windows on the desktop PC, or Internet Explorer over Linux on the laptop), however we choose reasonable settings, that are hopefully representative of both households and enterprises.

Conversely, closest related work such as [4] considers a large number of Websites (about N_W=100) but only a single HW/OS platform (for a total of N_P=4 settings), while [5] instead considers a large number of browser appliances (7 browsers on 3 HW/OS platforms for N_P=21) but a small number of Websites (N_W=3 technical blogs). Thus, while each of the above work performs a detailed analysis concerning a *single* aspect (e.g., SW or Web in our notation), they instead miss the overall picture: i.e., the *relative importance* of each factor, that is explored in the rest of this paper.

A. Platform Heterogeneity

Fig. 2, reports a few preliminary results to show the differences that arise due to HW, OS, and SW configuration. To estimate an *upper-bound* of the potential differences, we perform a stress test by simultaneously playing 5 high-quality

Fig. 1. Platform heterogeneity: specifications and configurations of OS, HW and SW.

TABLE I
WEB HETEROGENEITY: SET OF WEBSITES UNDER TEST.

Website	Alexa Rank	Content Type	No. of Flash Plugins
Gmail	1	Email Client	0
Yahoo	4	Online Portal	3
YouTube	3	Video Streaming	1
Wikipedia	6	Info. Portal	0
BBC	40	News Portal	2
CNN	58	News Portal	3
Gazzetta	585	Sport Portal	7
LeMonde	769	News Portal	6
Amazon	19	eCommerce	1
Fnac	1699	eCommerce	1
Slashfilm	4403	Review Blog	5
Mashable	292	Tech. Blog	4
Spotify	2834	Music Streaming	1
Collegehumor	1251	Entertainment Blog	1

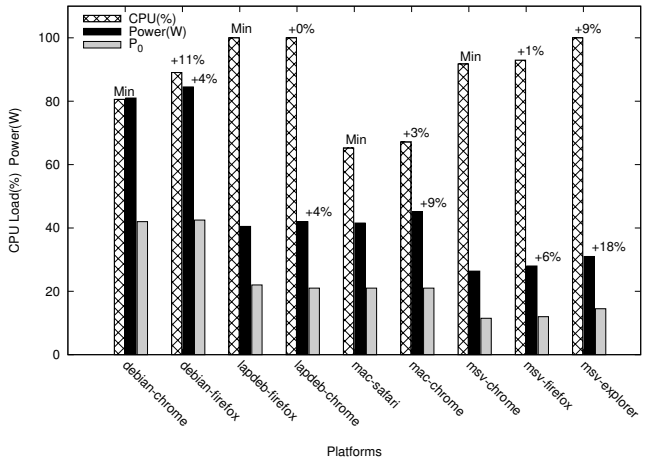

Fig. 2. HW, OS, SW Platform heterogeneity at a glance: CPU load and Power-consumption for stress-test benchmark.

videos from YouTube. Each test is repeated twice, cleaning the browser cache prior to each execution, so that content download is also taken into account.

During each test, which lasts for 120 seconds, we measure every 10 second the power-consumption P with a wattmeter and the CPU load L. Beside the browser under test, only a Task Manager (Windows) and a shell (OS X and Linux) processes were running (to measure L).

Fig. 2 reports the average CPU load L during the stress test with hatched bars, whereas the corresponding average power-consumption P is represented with dark solid bars. For reference purpose, we also report the power usage P_0 of the idle system, without any browser, with lightgray bars. Results are arranged so that, for each HW+OS configuration, the most power-efficient browser appears first: on top of each bar, we also report the percentage increase w.r.t the baseline browser.

Notice that while CPU load varies in the 70-100% range for all platforms, the power-consumption range (25-85 W) is wider due to the presence of a desktop PC (whose external LCD screen is not taken into account, as its plug is not measured by the wattmeter). The CPU load may saturate to 100%: at the same time, we see that different OSs for a given HW (e.g., Linux vs OS X for the MacBook HW) or different SW for a given OS (e.g., chrome or Firefox for Windows) may bring the CPU load below 100%. As the power-consumption may rise much above the baseline P_0, we already see that there is room for substantial power-savings.

B. Website heterogeneity

Due to the size of the Web, it is difficult to select a set of Webpages truly representative of an average use. For instance, [4] considers the 100 most popular Websites according to the Alexa ranking, out of which it then selects the 10 most CPU hungry sites. At the same time, though Google is the most popular site according to the ranking, (i) its homepage is lightweight by design, (ii) due to country-local replicas, is present more than once in the ranking and (iii) it likely represents a first quick hop toward other, more content-rich and thus CPU-intensive, Webpages.

We thus resort to a careful selection of Websites (reported in Tab. I along with their category, their Alexa ranking and the corresponding number of Flash plugins). The selected sites pertain to different services (e.g., blog, news, video, music, etc.), in an attempt to mimic different user activities. Notice that while some of the Websites are not world-wide popular,

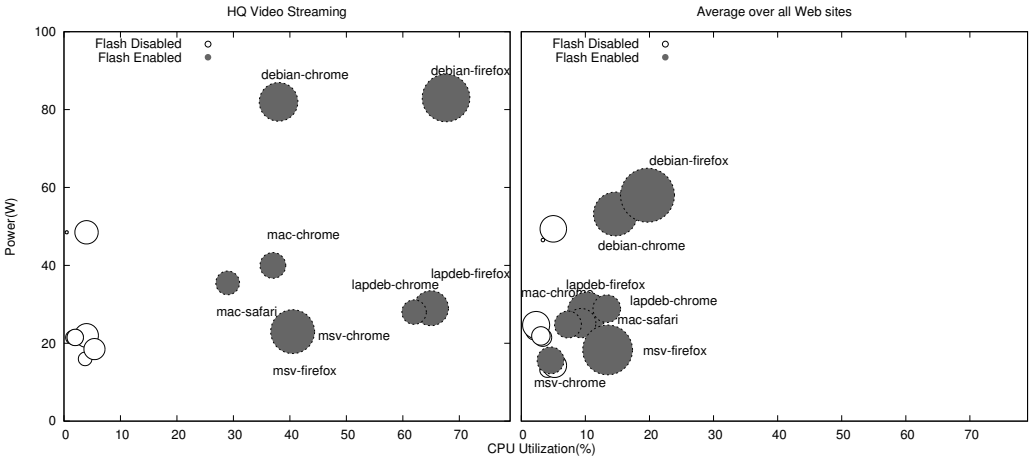

Fig. 3. Website heterogeneity at a glance: scatter plot of CPU load and Power-consumption for YouTube video (left) and averaged over the whole Webpage set (right). Circle width equals the standard deviation of the CPU load.

they are nevertheless very popular in the respective countries (e.g., Gazzetta in Italy, LeMonde/Fnac in France, etc.). Also, we carefully select pages containing a rather different number of Flash plugins (i.e., from 0 for Gmail/Wikipedia to 7 for Gazzetta).

For each of the considered Websites, we access (one at a time) the page with the different HW, OS, SW configurations earlier outlined, recording the CPU and power metrics during 120 seconds. In all our experiments, we either use Flash Player 10.0 (which was the most recent version of the Flash player at the time when the experiments were run[1]) or perform the same experiment with Flash disabled.

Results are reported in Fig. 3: in the form of scatter plot, each HW, OS, SW configuration is depicted as a circle, whose center is located at the average CPU load and power-consumption (L,P), and whose radius equals the standard deviation of the CPU load. Due to the importance of video (which is currently rising faster than any other type of service and is forecasted to account for over 90% of Internet traffic in the next few years [9]), the plot on the left only refers to YouTube, while the average performance over all sites of Tab. I is reported on the right. White circles represent the SW performance without Flash, whereas gray circles correspond to Flash enabled performance.

From Fig. 3, is easy to grasp that HQ video streaming is the most CPU hungry Web application, reaching up to 70% CPU utilization, while the average CPU load of other services do not exceed 25%. Also, the CPU rise depends on the specific configuration, as can be clearly seen from the dispersion of the gray circles with respect to the clustered white ones. Indeed, while the absolute power-consumption figure depends primarily on the HW platform, nevertheless differences may also arise depending on the specific OS and SW configuration.

[1]Notice that results may differ with the use of Flash Player 10.1 [8], especially in terms of *CPU load*. In fact, the new release is designed to exploit hardware acceleration capabilities offered by the graphic processing unit (GPU), by transferring the load from the CPU to the GPU. We however expect results to remain valid in terms of *overall* load (CPU+GPU).

III. Aftermath

We now assess how much the Web could be greened from the end-user point of view, considering two important aspects. On the one hand, we *relatively weigh* the impact of Platform (i.e., HW, OS, SW) and Websites on the overall end-user power-consumption. On the other hand, we conservatively quantify the potential *absolute power saving* that could be achieved with the simplest technique.

A. Relative weight

To gauge the relative importance of the considered parameters, we measure the extent of load L (or power P) variability by evaluating the ratio between the maximum and minimum L (or P) values gathered by varying a single parameter at a time, evaluating then the average over all other parameters.

For instance, to measure the Website impact on power variability I_P, we average over all possible platform settings $s = (HW, OS, SW)$ the maximum to minimum power-ratio $I_P = E_s[max_w P(s)/min_w P(s)]$ gathered over all Websites w experiments, where $E_s[\cdot]$ denotes the average over all possible setting $s = (HW, OS, SW)$.

Notice that, as our testbed does not exhaustively explore all HW×OS×SW combinations, the relative importance for some of the parameters is evaluated only on the representative subset. For the sake of clarity, to measure the HW impact we consider the desktop PC and the Mac laptop, with the Chrome and Firefox browsers under Debian, averaged over all Websites: as the desktop PC consumes more than the laptop, we have $I_P = E_w[\frac{1}{2}\frac{P(Desktop,Debian,Chrome)}{P(MacBook,Debian,Chrome)} + \frac{1}{2}\frac{P(Desktop,Debian,Firefox)}{P(MacBook,Debian,Firefox)}]$, where $E_w[\cdot]$ denotes the average over all Websites.

Ratios are reported in Tab. II for both CPU load L and power-consumption P. From the table, it emerges that OS impact is less preeminent than other factors. As expected, HW is the most important factor and may also induce a larger

TABLE II
RELATIVE IMPORTANCE OF FACTORS UNDER STUDY.

	I_L	I_P
OS	1.10	1.11
HW	1.45	1.95
SW	1.72	1.08
Web	4.63	1.44

variability of CPU utilization. Interestingly, the use of different SW browsers affects CPU usage, though not enough to bring significant changes to power-consumption[2]: as the power-ratio is lowest among all factors, this means that users can stick to their preferred browser without harming the planet too much.

Finally, it can be seen that the specific Webpage can have a dramatic impact on the CPU load, as this in turn translates into a large variability in the power-consumption: blame in this case should go to both developers of Flash scripts (who carelessly implement unoptimized scripts where an animated image would probably have sufficed) and to developers of the Flash interpreter (whose performance are far from being optimal).

B. Absolute power saving

Finally, we quantify the absolute power waste (or, potential gain). From our experimental study, it emerges that a single Webpage consumes on average about 4.7 W, which can grow up to nearly 16 W in the not-so-uncommon case of video streaming[3].

Users, while browsing, generally leave a number of tabs open, and may then close a batch of tabs from time to time. Indeed, unless the number of tabs causes a noticeable performance slowdown (in which case users would be compelled to close some older tabs), users have no real incentive in keeping only one or few tabs open at the same time. Yet, as users are likely consulting only a single tab, this also means that *all but one tabs are unnecessarily using CPU, hence wasting power*.

A simple solution to this waste, inspired by grandmother precepts and common sense, could be summarized as "*Do not leave the light on if you are in another room*". This simple statement, applied to the world of Web browsers, could be better rephrased as "*run only the scripts on the currently active tab, of the focused window, on the focused desktop*". Within the browser, this means freezing the execution of all scripts which are not on the focused tab (e.g., implemented as sleep/resume signals sent to the script interpreter on each tab focus change). Further gain could be obtained in the case browsers stopped the execution of scripts in tabs that are hidden to the end-user (e.g., tab is focused but the corresponding window is iconized, or in another non-visible desktop, etc.).

To evaluate the *lower bound* of the power-saving for this simple strategy, we can consider that users have only two tabs open, one which they are actively consulting, while the other is open but not focused. In this case, given the estimated average consumption per Webpage, reduction would be on the order of a 10 W light bulb. To gather an *upper bound* of the power-saving, it would be necessary to estimate the number of tabs that could be left open without causing a noticeable slowdown: although the exact CPU threshold after which the slowdown becomes noticeable is rather subjective, bounding CPU utilization to 75-90%, our measurement suggests that at least 7-10 tabs could be opened in parallel without noticeable performance drop even for the slowest platform. Considering a household of 3 people, it is not unlikely that the waste could turn into a 100 W power-drain: as countries around the world are in the process of phasing out 100 W light bulbs in favor of low-energy alternative, it seems that Web browsing should be ready to undergo the same revolution as well.

[2]With minor exceptions, results in [5] point toward similar conclusions.

[3]Estimates are consistent, although higher, with respect to the 3.3 W-11.3 W range given in [4] for a single platform

IV. CONCLUSIONS

Web browsing has evolved much, concerning the richness of its content, the type of offered services and the available user interfaces. Considering client-side dynamically generated content, we argue that another evolution is around the corner: i.e., *the reduction of unnecessary power-consumption.*

To provide an initial assessment of this issue, we perform a thorough measurement campaign, considering several factors such as hardware platforms, operating system, browser applications and Websites. Our results show that Webpage content is, after the specific hardware platform, the primary factors affecting Web browsing power-consumption. Experiments estimate the average power-consumption of a Webpage to about 5 W, which can grow up to nearly 16 W in the not-so-uncommon case of HQ video streaming.

We argue that a potentially large source of power waste arises due to tabbed browsing, causing several scripts to run in parallel: yet, as users typically interact with only one tab at any given time, scripts running on unfocused tabs unnecessarily waste resources. Clearly, the presented results are far from being exhaustive – as this would require considering many more platforms configurations and Websites. Nevertheless, we believe this work to be a valuable starting point, that also outlines a simple yet effective way for a greener Web browsing.

ACKNOWLEDGEMENTS

The authors wish to thank their granmas, whose useful precepts go well beyond the scope of this paper.

REFERENCES

[1] "Flash player version penetration." http://www.adobe.com/products/player_census/flashplayer/version_penetration.html.
[2] "Comparing embedded live stream options." http://streaming.wisconsin.edu/streaming_comparisons/live/flash_winmedia.html.
[3] J. Ozer, "Flash player: Cpu hog or hot tamale? it depends.." http://www.streaminglearningcenter.com/articles/flash-player-cpu-hog-or-hot-tamale-it-depends-.html.
[4] R. Hansen, "Browser power consumption." http://www.sectheory.com/browser-power-consumption.htm.
[5] "Browser face-off: Battery life explored." http://www.anandtech.com/show/2834/2.

[6] X. Fan, W.-D. Weber, and L. A. Barroso, "Power Provisioning for a Warehouse-sized Computer," in *Proceedings of the 34th Annual ACM International Symposium on Computer architecture (ISCA 2007)*, (San Diego, California, USA), pp. 13–23, June 2007.

[7] J. Baliga, R. Ayre, W. V. Sorin, K. Hinton, and R. S. Tucker, "Energy Consumption in Optical IP Networks," *Journal of Lightwave Technology*, vol. 27, pp. 2391–2403, July 2009.

[8] "Cpu performance test results: Flash player vs. html5." http://blogs.adobe.com/flashplatform/2010/03/cpu_performance_test_results_f.html.

[9] "Cisco visual networking index: Forecast and methodology, 2008-2013." http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/-ns705/ns827/white_paper_c11-481360_ns827_Networking_Solutions-_White_Paper.html, June 2009.