STUDIA INFORMATICA 2013
Volume 34 Number 2B (112)

Artur OPALIŃSKI
Politechnika Gdańska, Wydział Elektrotechniki i Automatyki

IMPROVING WEB USER EXPERIENCE WITH CACHING USER INTERFACE

Abstract. In human-computer interaction, response time is assumed generally not to exceed significantly 1-2 seconds. While the natural competition in the Internet public Web serving ensures adhering widely to such limits, some Web environments are less competitive and offer much worse user experience in terms of response time. This paper describes a solution to significantly improve user experience in terms of response time with only modification to the user interface, by caching key pieces of data needed for quick browsing, during some tasks performed by humans.

Keywords: user experience, response time, cache, data prefetching, Web User Interface

POPRAWA ERGONOMII INTERFEJSU UŻYTKOWNIKA WEB POPRZEZ STOSOWANIE CACHE

Streszczenie. Pożądany czas odpowiedzi systemu komputerowego przyjmuje się często w przedziale do 1-2 sekund. Konkurencja wymusza zachowanie tego czasu przez organizacje prowadzące publiczne strony WWW. Techniki Web stosowane są jednak także jako część większych rozwiązań czy złożonych produktów, gdzie konkurencja nie wpływa tak bezpośrednio na utrzymywanie krótkiego czasu odpowiedzi i jest on często dłuższy. W artykule opisano rozwiązanie polegające na stworzeniu dodatkowego interfejsu użytkownika z cache, który pozwala znacząco skrócić czas odpowiedzi dla predefiniowanych zadań użytkownika.

Słowa kluczowe: ergonomia, czas odpowiedzi, pamięć podręczna, wstępne pobieranie danych, webowy interfejs użytkownika

1. Introduction

In human-computer interaction, response time for undisturbed work is assumed differently, depending on studies and user tasks, but generally not significantly exceeding 1-2 seconds [1-5]. The natural competition among content or services providers on the World Wide Web ensures that Internet sites strive to adhere to this limitation by employing various techniques and technologies, including, but not limited to HTTP cache-control settings [6]. It is an important duty of Web sites developers and maintainers to ensure site usability, including low response time. Increase in response time is directly attributable to decrease in customer satisfaction [4, 5].

Web technologies are also applied for other tasks than public Web serving. Often, Web technologies are used to operate or to configure network-enabled equipment, to configure and administer modular applications, or as teaching environments. The comfort of human work requires a similar response time in these applications as in the Internet. Unfortunately, no direct competition drives here the response time low: the user is not free to operate device through another interface if the Web interface is too slow, and is not free to switch to quicker learning environment than that provided to him by his educator as part of the whole learning process. Surely the solutions providers, which include the Web component into their products or services, also regard the response time, but this single parameter is not paramount for their complete solutions. This causes less pressure on developers and maintainers to keep response time low.

Therefore using Web technologies for applications other than public Web browsing may lead to worse user experience. A practical example which directly lead to this research work was the upgrade of the Moodle e-Learning environment to version 2.x. The new environment has introduced serious delays in human-computer interaction, due to increased complexity of both the Web Application as well as the HTML pages parsed and rendered by the browser.

The response time acceptable by users depends on many factors, but it becomes especially important when users need to quickly switch forth and back between pages, just looking over the content, and jumping to other places. The Web page load time is influenced by:

- Web application processing time,
- network transport time, influenced by bandwidth and latency, but also by number of page components resulting in separate HTTP GET requests,
- page Document Object Model (DOM) processing time,
- time for rendering page on screen.

To improve response time, various forms of caching at different levels are employed. Caching efficiency is limited by cache size, and additionally influenced by the ability to pre-

dict future data references (usually based on spatial or temporal locality of reference) [7, 8]. When data is less voluminous than cache size, all data can be cached in advance and served from the cache.

But it is hard to integrate such cache into standard Web application architecture. First, if the cache is to be pre-loaded (warmed), the control signal to do so at some point is needed. Then the question of cache localization arises; of course the most efficient cache should possibly be located close to the Web browser. Both these questions lead to outcome, that such cache should be integrated into Web browser. Such task is non-trivial and has been abandoned in this work.

Fortunately, the HTML page structure and content of the product or solution employing Web technologies are relatively stable, compared to public Web pages. Thus the Web User Interface is either static and controllable (when this it is a local application) or at least not changing frequently (as it is with free external Web site services where valuable maintenance is reduced).

Also, after prolonged time of usage, the most daunting and time-consuming user tasks are predictable and the user behavior pattern is known in advance. Additionally, in teaching environments, the user behavior pattern is often very common among users at a specific time of semester. E.g. there are specific parts of content which are revisited massively before summative quizes, or after semester ends.

The goal of the research described in this paper is to significantly improve user experience in terms of response time with only modification to UI, by caching key pieces of data needed for quick browsing, as typical during quick reading, or visually comparing data. The solution os tailored to a specific Web application and to a subset of application's HTML content.

2. Solution

To improve the user experience in regard to response time when performing specific tasks, a dedicated caching user interface (CUI), layered parallel to the Web User Interface (WUI), has been introduced (Fig.1). This interface is by design not to replace the WUI, but it is dedicated to perform just one or few pre-defined task, which are page-load-intensive (browsing, comparing short texts, analyzing information).

The CUI offers the users simple controls to start the load intensive tasks to warm the local cache. The cache is filled with data originated from the Web Application, by issuing HTTP GET requests defined by the specific task. The scope of the GET requests, i.e. the scope of the data to cache, is easily configurable. After loading the cache, another set of controls al-

lows for quick random access to predefined parts of the data, for browsing, combining, or analyzing it.

The tool has been specifically designed for the sake of grading student activity on an e-Learning course web forum. Data is cached in a local SQL relational database, using the engine provided by SQLite[9] library. The database diagram is presented in Fig.2.

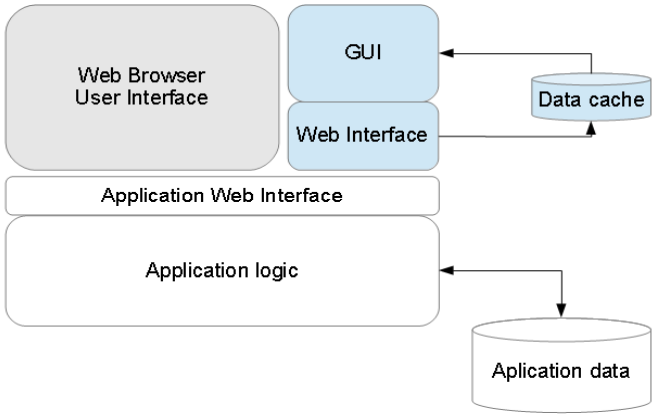

Fig. 1. Solution architecture
Rys. 1. Architektura rozwiązania

Using the standard WUI, the user needs to perform the following tasks to grade students' posts:

- load the login page and authenticate,
- load the course main page and locate the forums (there are separate forums on the e-Learning course for each homework assignment)
- load the page with student replies,
- identify the posts of specific students (e.g. students without grades),
- during grading, compare the reply to this student replies and other students replies, to the discussion thread original topic, and refer to comments added to already graded student posts.

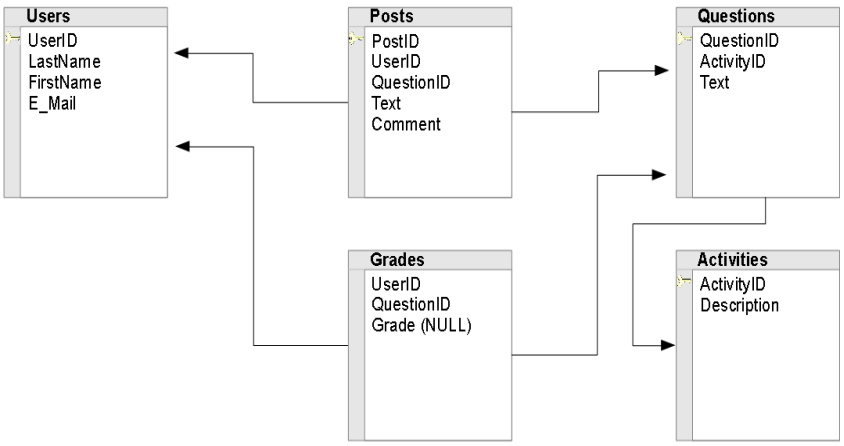

Fig. 2. Database diagram
Rys. 2. Diagram bazy danych

CUI (Fig.3) written with wxWidgets library [10] allows to perform the same tasks, but time for page loading is not needed, as complete data is already in the pre-warmed cache. HTTP communication interface is provided by the libcurl [11] library.

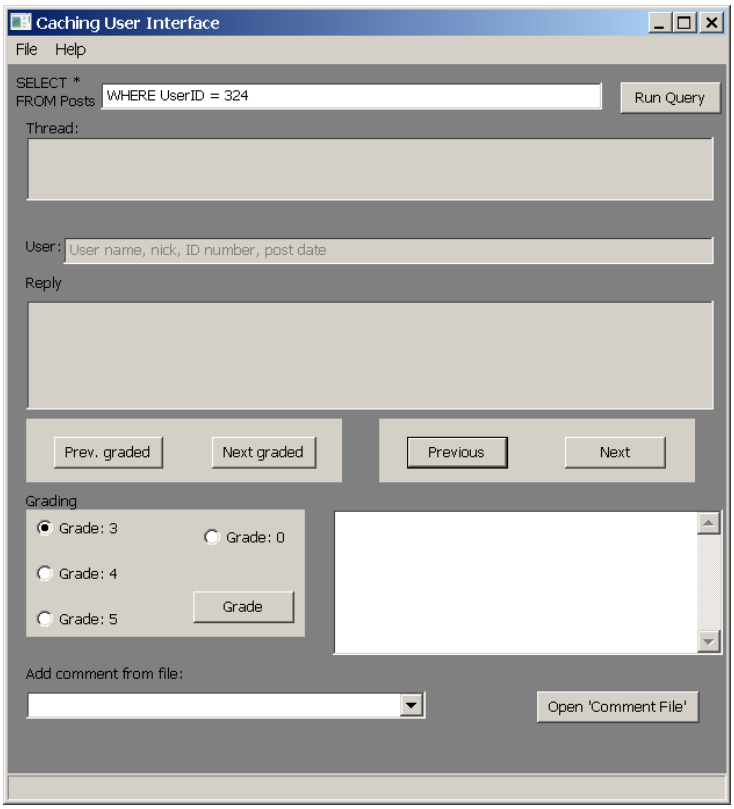

Fig. 3. User Interface
Rys. 3. Interfejs użytkownika

3. Verification

The response time over WUI has been evaluated on 36 hosts, in 4 rooms, connected to 3 IP networks. Two of the networks were equipped with Web proxies. To evaluate the response time, a short shell script has been run, calling wget[12], a program for non-interactive retriving of files using HTTP protocol. The measurements were conducted consecutively on all hosts, over a continuous period of approximately 3 hours. The shell script performed authentication, and requested 10 consecutive assignment sub-pages from the course main page, and then the next sub-page (discussion thread page) on each assignment page. Therefore each script run involved retrieving 20 different pages after authentication. The script run did not involve any parsing of HTML content, so the run time of the script mainly encompasses response time during 20 page loads, plus response time needed for the initial authentication step.

The obtained response time during single page load is presented in Fig.4. The average response time per single page load is above 7 seconds, which is well above the widely accepted human limit, mentioned in the Introduction.

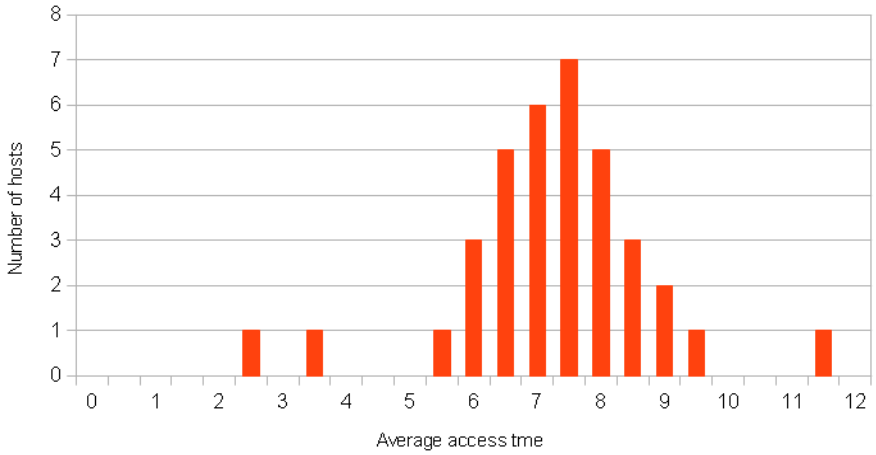

Fig. 4. Average single page load time distribution
Rys. 4. Rozkład średniego czasu ładowania pojedynczej strony

The host with the shortest response time of 2.5 seconds has been selected for further verification. The CUI has been installed on it, and cache warming started. Average cache warming time in 10 consecutive attempts was 62 seconds. This time in each attempt included non-interactive authentication, non-interactive parsing of HTML course and assignment pages, and non-interactive retrieval of the same HTML pages as in the WUI evaluation. The cache warming took a similar time as the WUI test on this host, as the overhead of HTML parsing was negligible related to the page retrieval time.

With the discussion threads data cached, all posts of a single student over all activities (all cached discussion threads) have been selected. To provide enough data, an active student has been chosen, with posts in each discussion thread (10 threads). Using the CUI, and switching by hand to next posts, took 18 secs. In this time, information from 10 different pages, accessible through additional 10 intermediate HTML pages, has been accessed. This is equivalent to access-time per page of 0.09 second. This result is significantly lower (better) than with standard WUI and allows for fully interactive and non-disturbed user work, in the scope hard-coded in the CUI.

No standard functionality has been jeopardized, as the whole Web application functionality remains continuously available through the standard WUI.

4. Conclusion

To allow for interactive work with remote Web application which cause otherwise unacceptable response time, a solution involving Caching User Interface (CUI) has been presented. The CUI is unchallenged in respect to response-time, but it only offers limited functionality compared to Web User Interface (WUI).

CUI only encompasses predefined usage cases. Extending it to more usage cases is entirely feasible, but then replicating even complete site data does not solve the problem of lacking local infrastructure (e.g. no Web Server). Therefore extending CUI functionality behind some limits would require changes to the Web application itself.

CUI compares well to browsing in tabs. First, not all Web applications support tabs. Besides, tabs do not provide synchronized scrolling of the pages already loaded, nor searching in concert on the set of loaded pages. Therefore tabs can be seen at most to address the load page time, not to provide smooth access to content.

Cache warming in CUI may take considerable time, depending on the amount of data and transfer speed. A long cache warming time makes CUI inefficient if only small amount of information is required. WUI should be used in such cases. In some applications – e.g. in e-Learning environments, cache warming can be limited to just one synchronization. A lot of content on e-Learning sites remains static after the process of its creation is finished. This pertains to both material provided by the teacher, which appears and accumulates till course end, as well as to student work – which also appears over the semester and only accumulates till semester end. Most searching over the pages is needed at semester end – when the complete data is available. So just one synchronization can be done, or a series of synchronizations can be performed at specific instants during the semester.

BIBLIOGRAPHY

1. Fui-Hoon Nah F.: A study on tolerable waiting time: how long are Web users willing to wait? Behaviour & Information Technology, Vol. 23, Issue 3, 2004.
2. Nielsen J., Molich R., Snyder C., Farrell S.: E-Commerce User Experience. Nielsen Norman Group, Fremont, CA 2001.
3. Artz D.: The Secret Weapons of The AOL Optimization Team. Velocity Conference, 2009.
4. Simic B.: The Performance of Web Applications: Customers Are Won or Lost in One Second. Aberdeen Group, 2008.

5. Forrester Consulting: eCommerce Web Site Performance Today: An Updated Look At Consumer Reaction To A Poor Online Shopping Experience. A commissioned study conducted on behalf of Akamai Technologies, Inc., 2009.
6. Fielding R., Gettys J., Mogul J., Frystyk H., Masinter L., Leach P., Berners-Lee T.: RFC2616, Hypertext Transfer Protocol – HTTP/1.1, sections: 13 Caching in HTTP and 14 Header Field Definitions. Network Working Group Request For Comments, 1999.
7. González A., Aliagas C., Valero M.: A Data Cache with Multiple Caching Strategies Tuned to Different Types of Locality. ACM International Conference on Supercomputing, 1995, p. 338÷347.
8. Byna S., Chen Y., Sun X.-H.: A Taxonomy of Data Prefetching Mechanisms. I-SPAN, IEEE Computer Society, Int. Symposium on Parallel Architectures, Algorithms, and Networks, Sydney 2008.
9. SQLite database engine software library Home Page, http://www.sqlite.org/ (15.01.2013).
10. wxWidgets GUI library Home Page, http://www.wxwidgets.org/ (15.01.2013).
11. libcurl client-side URL transfer library Home Page, http://curl.haxx.se/libcurl/ (15.01.2013).
12. GNU wget software package for retrieving files, Home Page, http://www.gnu.org/software/wget/ (15.01.2013).

Wpłynęło do Redakcji 16 stycznia 2013 r.

Omówienie

Pożądany czas odpowiedzi systemu komputerowego przyjmuje się często, zależnie od szczegółowego kierunku badań oraz zadania wykonywanego przez użytkownika, w przedziale do 1-2 sekund [1]. Taki czas jest kontrolowany i zachowywany przez organizacje prowadzące publiczne strony WWW, gdyż wymusza to naturalna konkurencja związana z walką o klienta [4]. Techniki Web stosowane są jednak także jako część większych rozwiązań czy złożonych produktów, między innymi w środowiskach e-Learning do wspomagania nauczania na odległość. Użytkownik środowiska e-Learning, korzystający z kompletnej oferty edukacyjnej, nie ma już swobody wyboru ośrodka o krótkim czasie odpowiedzi, zaś osoby nadzorujące proces nauczania nie kierują się wyłącznie długością czasu odpowiedzi. Stąd w niektórych zastosowaniach techniki Web, czas odpowiedzi nie jest niski, gdyż nie pojawia się presja bezpośredniej konkurencji.

Artykuł proponuje uzyskiwanie znaczącego skrócenia czasu odpowiedzi przy wykorzystaniu portalu Web do e-Learningu, poprzez stosowanie w predefiniowanych sytuacjach dodatkowego interfejsu użytkownika z cache. Interfejs taki musi być każdorazowo dostosowany do funkcjonalnego zakresu aplikacji Web. Przedstawione zostało przykładowe rozwiązanie (rys. 1), obejmujące interfejs użytkownika, interfejs Web oraz lokalną bazę danych SQL. Rozwiązanie to jest wykorzystywane do oceny aktywności studentów na forum dyskusyjnym w kursie prowadzonym z wykorzystaniem technik e-Learningu. Podczas oceny aktywności studentów na forum dyskusyjnym konieczne jest przeglądanie wielu wpisów (postów) i wątków dyskusyjnych, z możliwością powrotu i porównania prezentowanych myśli. Narzuca to szczególne wymaganie na niski czas odpowiedzi podczas przeglądania wpisów. Zaproponowane rozwiązanie pozwoliło uzyskać skrócenie czasu odpowiedzi z około 2,5 sekundy do poniżej 0,1 sekundy, a zatem poniżej progu uznawanego za zauważalny dla człowieka.

Address

Artur OPALIŃSKI: Politechnika Gdańska, Wydział Elektrotechniki i Automatyki,
ul. Narutowicza 11/12, 80-233 Gdańsk, Polska, Artur.Opalinski@pg.gda.pl.