Best Practices to Increase the Performance of Web-Based Applications

B. Alkhatib[1], M. Kawas[2]

Abstract – *In today's world, technology development aspects are growing rapidly in a way that makes the development of these aspects unacceptable to slowdown. Perhaps the most distinguishing aspect is the World Wide Web which is considered as the main container for these prospects, despite of the challenges and the difficulties which have faced it since the beginning especially in term of performance. This paper states current performance difficulties that face web-based applications, grouping these difficulties into categories based on the web technology used.*

It also proposes a number of recommendations and enhancements that increase and optimize the web performance. These recommendations are implemented in a real case which is "Mofadalah" web-based application at the Syrian ministry of health. ***Copyright © 2013 Praise Worthy Prize S.r.l. - All rights reserved.***

***Keywords**: Web-Based Applications, Performance, HTML, CSS, JavaScript, Ajax*

Nomenclature

HTTP	Hypertext Transfer Protocol
CSS	Cascading Style Sheets
HTML	Hypertext Markup Language
XML	Extensible Markup Language
JSON	Javascript Object Notation
AJAX	Asynchronous JavaScript and XML
AJAH	Active JavaScript and HTML
RSS	Rich Site Summary
API	Application Programming Interface
URL	Uniform Resource Locator
fps	Frames Per Second

I. Introduction

Transferring contents in an easy way and speeding up the process of displaying them are considered as the most determinants of the web performance optimization, where speeding up the process of displaying content is the most important factor for websites to success, attract more user and increase profits.

Furthermore, users that surf the internet with high speed connections are unable to wait more than six seconds for a page to open due to the high prevalence of the broadband [1]. Therefore, the programmer has only three to four seconds to display the whole web page content; otherwise he could lose visitors of his website.

Nowadays, any web page contains more than fifty external objects in average, and these objects need to be defined at the beginning of that web page using Object Overhead which affects the response time.

II. Lack of Performance Issues

II.1. Customer Dissatisfaction

Users will not be satisfied when a requested web page needs more than 10 seconds to load, whereas, Broadband users cannot sustain slow loading as Narrowband users.

A survey had been published in "JupiterResearch" website, which found that 33% of users that had worked on Broadband were not happy about the idea of waiting more than 4 seconds for a web page to load.

In contrast 43% of users that had worked on Narrowband were not even ready to wait more than 6 seconds [2].

II.2. Negative Impact on Companies' Profits

Simple changes on time required to load a webpage or on server's responding time to serve website visitor's requests have a significant impact on companies' profits.

According to Google Research Center which showed that a web page of 10 search results takes a period of 0.4 of a second to load while a web page of 30 search results takes 0.9 of a second, which would negatively affects the business advertising profits by 20 percent [3].

Furthermore, Google Research Center showed that reducing the size of Google Maps homepage from 100 Kb to 70-80 Kb had a positive impact on the profits of business advertising of 10 percent during the first week and 25 percent after three weeks [4].

In addition, a series of tests on Amazon shown similar results, for each 100 millisecond of webpage loading time, sales process would be reduced by 1% [5].

Manuscript received and revised June 2013, accepted July 2013

II.3. Number of Visitors

Web pages that either have a slow loading or a slow response time will reduce the credibility and the quality of their website and will also negatively affect users' feedback. Actually, there is a strong correlation between the number of visitors of a website and its response and loading time. Response and loading time is the most important factor after the design a website which attracts users [6].

II.4. Visitors' First Impression

In general, a visitor takes a quick first impression for a website. This impression takes an average of 1/20 of a second to be formed and it will stay the same no matter what he visits on the website, or how many times he visits it. Furthermore, studies had conducted by Noam Tractinsky showed that the average of the attractiveness of a webpage after loading for half a second is equal to the average of it after 10 more loading seconds.

Therefore, programmer of a website has a very short time to get a positive impression for his website from its visitors [7].

II.5. Webpage Size

The size of any webpage has increased significantly over the past years, where the average of this increasing since 2003 is more than 3.3 times, which means the average size of a webpage has increased from 93.7 KB to more than 312 KB. Over the same period, objects and added tags of a webpage, such as additional files, images, cascading style sheet and JavaScript, have increased by approximately 2 times (25.7 to 49.9) [8]-[9].

Older web pages contained a small number of objects; therefore, the size of the body was the primary factor of the response time. However, with the increasing of the number of objects in a single webpage, the delay in response time became related to the header section (not the body itself). This fact is considered as a main issue in the response delay problem [10].

II.6. Maximum Response Time

Due to the size increasing of a webpage, visitors that work on Narrowband (56 KB, ISDN) suffer from slow response, in contrast, visitors that work on Broadband surf the internet with a quicker response. Keynote Company recommends using the following theory for the response time of websites on networks with variety of speed:

- 2 to 3 seconds for high speed networks used particularly for companies.
- 3 to 5 seconds for DSL networks.
- 20 to 30 seconds for dial up phone networks.

The above recommendations are applicable to main web pages or homepages of a website where visitors usually should navigate quickly. But there are some exceptions, for pages that visitors take time while browsing it, such as news page, because of the attention of a visitor for a specific subject may let him sustains the performance issue of a page.

III. Proposed Improvements to Enhance the Performance of Web-Based Applications

There are a lot of suggestions and modifications that can be applied to a web-based application to improve its performance.

III.1. Webpage Recommendations

III.1.1. Reduce HTTP Requests

HTTP requests are considered as a primary factor that affects webpage speed, so the number of server requests must be reduced which requires reducing the number of external objects in the webpage, because each one of these object is an HTTP request.

Increasing the number of external objects will delay page loading and when their number becomes more than four, their loading process will start first even before starting webpage's main content loading process.

The following recommendations can reduce HTTP requests:

Convert Images to Text with Styling

Images are usually used for page titles or for menus styling which could be replaced using CSS (Cascading Style Sheets), and search engines do not index the text in images so they send unnecessary requests when indexing such webpage.

Use Text Overlays

Sometimes, programmers have to use images that contain text which increase the size of these images, and to avoid this size increasing we must isolate the image from the text that is associated with and use CSS files to get the required style which accelerate server response.

Convert Spacing Between Cells to Margin or Padding

Programmers used to use small-dimensional images to add spaces between elements. This should be replaced using margin or padding in CSS files.

Merge Images and Use Sprite Instruction

Programmers can use Image map or Sprite instruction after merging multiple images in a webpage to reduce the number if HTTP requests into one request.

Merge and Improve CSS and JavaScript

Many programmers create several CSS or JavaScript files for their website. This approach will result additional requests. Thus these files should be merged into one.

Link CSS or JavaScript Files

This method reduces the number of HTTP requests by integrating external CSS and JavaScript files automatically by linking them on server, this approach is called Suturing.

Store and Hide the Dynamic Files in Cache

Programmers could add a particular script at a webpage header to cache their external files and this will reduce the number of HTTP requests in future calls.

Remove (i)Frames

(i)Frames are used in more than 52% of web pages; they are used to display commercial ads or another webpage which cause a slow performance of these pages due to the additional required HTTP requests.

III.1.2. Resize and Enhance Images

Manufacturers of digital cameras aspire to increase the resolution of their images which cause an increasing in their size in an amount that is not suitable to be used in a webpage.

III.1.3. Multimedia Enhancement

Multimedia files require HTTP requests but they take a large part of network bandwidth. The following recommendations can enhance the process of embedding multimedia files into a webpage:

Enhance Videos to Work on the Web

To ensure that a video is appropriate for the web, it should have a short time, small dimensions and a proper codec [11].

The Rate and Dimensions of Video-Frames

Increasing frame rate, which means increasing the number of frames per second, will decrease the flickering and increase video's consistency, but on the other hand, it will produce more data. Studies have shown that videos with less than 12 to 15 fps have reduced the perception of users [12].

Increasing frame dimensions, which means increasing the number of pixels representing the frame, will increase video's quality, but on the other hand, it will also produce more data. Frame dimensions should be more than 320 x 240 pixels.

Reduce Noise and Movement

This can be accomplished by:

- Minimize the movement of the camera using a sitter for the camera if that's possible.
- Minimize the movement of the photographed objects.
- Use an intensive light.
- Use a simple background.
- Avoid zooming in or out and lens rotations.
- Use professional tools.
- Use digital formats and forms.
- If the base or the sitter of the camera is not available, an equilibrium tool or lenses for steady images can be used instead.

Edit Video

Videos can be edited to meet web standards and requirements, and these are the most important edits that should be considered:

- Reduce dimensions to meet web standards.
- Use the minimal frame rate, if possible.
- Cut unclear edges from the pictures.
- Reduce video's noise using filters.
- Modify color contrast.
- Modify gamma constant for cross-platform variation.
- Restore black and white.
- De-interlace.
- Choose a proper codec.

III.1.4. Convert JavaScript Code to CSS

JavaScript codes are widely used in web with percentage of 84.8%. These files are used in many cases such as form validation, menus, mouse rollover, browser information detection, statistics and complex Ajax applications. Web programmers can achieve most of these functionalities with approximately the same quality and with more efficiency using CSS.

III.1.5. Use Sniffing Approaches on Server

JavaScript codes are used noticeably in browser sniffing to detect browser's type, browser's version number and its ability to support some instructions.

Therefore, to reduce JavaScript codes, the programmer should use some instructions to be performed on the server if possible.

III.1.6. Improve JavaScript Files' Size

After converting most of JavaScript code to CSS files and moving as much code as possible to be performed on the server, JavaScript code can be optimized to reduce files size. The programmer can use short names for objects, variables and functions to reduce the number of bytes used to store the code. Programmer can also use automatic engine to accomplish this process such as w3compiler which enhances scripts.

III.1.7. Move Table Layout to CSS

Using CSS files to style a webpage reduces a large amount of html code written to display that page with an average of 25% to 50%. Programmers should check if CSS and DIVs can be used to replace table layout code.

III.1.8. Write CSS Code into Separate Files:

This technique is the most important because it saves network bandwidth and also reduces website maintenance time as it separates the actual content of a webpage from its style. So programmers are advised to avoid using inline styles.

III.1.9. Keep Visitors Busy

Programmers can improve their websites by loading the useful and consistent contents of their web pages first to attract visitors and keep them busy. As such, Weather Underground website loads the weather status in the upper-left corner in its homepage at the beginning then completes the loading process.

III.1.10. Use JavaScript Code Wisely

External JavaScript files that are linked to a webpage at its header delays loading the actual content of that page.

III.2. CSS Recommendations

III.2.1. Group Selectors with Similar Declarations

CSS allows programmers to group multiple selectors that share similar statements separating them using commas between their names. This reduces the size of CSS files. For example:

```
 .sitehead {
       font-weight: normal; font-size: 12px;
}
.sitenav {
      font-weight: normal; font-size: 12px;
}
Becomes:
.sitehead, .sitenav {
      font-weight: normal; font-size: 12px;
}
```

III.2.2. Group Declaration with Similar Selectors

CSS allows programmers to group several style declarations into one shared selector separating them using semicolons. This also saves CSS files size. For example:

```
body {font-size: 1em;}
body {color: # 000;}
Becomes:
body {font-size: 1em; color: # 000;}
```

III.2.3. Merge Styles Using Classes

CSS allows programmers to group several declarations that share the same style into a certain class allowing him to use that style as much as he needs without repeating the styling code.

III.2.4. Use the Inheritance Property to Get Rid of Repeated Declarations

CSS allows programmers to inherit selectors' statements from their prior declarations according to tags hierarchy that use these selectors in a webpage.

III.2.5. Use CSS Shorthand Formulas

CSS allows programmers to reduce a lot of declarations using some supported abbreviations.

For example, instead of specifying color values using hexadecimal (rrggbb), programmers can use (rgb). Moreover there are a lot of declarations that can be abbreviated such as lines, edges, padding and margin.

III.2.6. Avoid Long Names for IDs and Classes

Long names for IDs and Classes are more understandable for programmers, but it loads additional bytes.

III.2.7. Use New Features of CSS

CSS version 2 and 3 provide an ability to specify elements with specific attributes using the following:

- [att]
- [att = val]
- [att ~= val]
- [att = val]

III.3. JavaScript Recommendations

III.3.1. Remove JavaScript Comments

It's very important to delete the comments in JavaScript files which are defined using // or /* */ symbols as these comments increase the size of JavaScript files and they do not provide any useful feature for the end user.

III.3.2. Conditional Comments

Programmers should beware of the conditional comments which are often used for a piece of code that is not supported by a particular browser as these comments also increase the size of JavaScript files. Moreover, using instructions and statements that are supported by all browsers is very important.

Conditional comments example:

```
<!--[if 1t IE7] <script src="patch.js"
type="text/javascript" ></script><![endif]-->
```

III.3.3. Reduce Whitespaces

JavaScript files may contain a lot of whitespaces so programmers are advised to remove as much of them as possible and without any negative impact on his website.

III.3.4. Use Abbreviated Instructions

Programmers should use the abbreviated instructions whenever it is possible. As in using x++ instead of x=x+1.

III.3.5. Use Constant Strings

Programmers should declare constant strings at the top of JavaScript code file to be able to use them later without reassign their values. This also reduces the size of JavaScript files.

III.3.6. Avoid Writing Instructions that Will Never Be Executed

Programmers must beware of the logical errors while writing their codes.

Such as, writing instructions inside conditional phrases or loops that are not reachable.

III.3.7. Reduce the Length of Variables and Functions Names

Programmers declare variables and functions that have clear names which will add additional bytes to the size of the code files.

III.3.8. Default Values

There are a lot of types and functions that have a default value so programmers should avoid the explicit assignments or declarations if they result the same as the default ones.

III.3.9. Use AJAH Instead of Using RSS, JSON or XML

AJAH is a type of asynchronous JavaScript and HTML codes that are used to get server responses formatted as HTML instead of JSON or XML to be viewed on a webpage directly without any further instructions.

This approach increases the speed of the process of viewing requested data from the server.

III.4. AJAX Recommendations

III.4.1. Use AJAX When Needed

Using AJAX could have negative impacts because even if requests from client to server have a small size but they may take a long time to get the response, thus it's better to avoid using AJAX as much as possible.

III.4.2. Use a Supported and Well-Designed AJAX Library

Programmers should use the following AJAX features:

- Ajax connections API.
- Easy access to all HTML nodes.
- Organize events.
- Visuals effects.

III.4.3. Reduce HTTP Requests

According to HTTP specifications, only two requests from a browser can be sent to a server simultaneously.

Thus, programmers must be aware of this problem and decrease HTTP requests as much as possible.

III.4.4. Choose Correct Data Types before Sending Them to Server

Even if AJAX requests have small size, but there are differences between the sizes of different data types so programmers should use the smallest data types that achieve the required task.

III.4.5. Provide a Cache Memory for JavaScript

Programmers may declare two arrays one for all AJAX requests and the other for the responding AJAX responses, so they can retrieve saved responses whenever they needed.

III.4.6. Make Sure of Handling All Network Performance Exceptions

Programmers should build an eighth layer above the HTTP seven layers [13], which is used to fix any problems may occur due to the lack of server responses to AJAX calls.

III.5. Server-Side Recommendations

III.5.1. Improve Parallel Downloads Methods

HTTP version 1.1 recommends that browsers should limit their requests to two simultaneous at most. Thus, programmers should use HTTP version 1.0 especially for images requests to allow parallel downloads feature.

III.5.2. Minimize DNS Lookups

DNS provides the ability to translate domain names into Internet Protocol (IP) address and vice versa and allows a website to be served by several servers; this process usually takes 20-120 milliseconds for each server lookup. Furthermore, browsers wait DNSs to finish their work before loading the website.

III.5.3. Compress HTTP

Compressing HTTP is a known approach used to decrease text content size which is transferred between servers. This approach uses well known compression algorithms such as GZip Compress that compresses all types of text files.

III.5.4. Delta Encryption

This approach is used to serve webpages by sending the differences between the sequenced versions of these pages. Server sends the page differences from the last request, which reduces the amount of transferred data in HTTP.

III.5.5. Rewrite Internet URLs using Mod_Rewrite

Programmers should use this technique to abbreviate

URIs over the internet, this approach will reduce few Bytes and website address will be optimized by search engines.

IV. "Mofadaleh" of Syrian Health Ministry

This is an integrated system used by Health Ministry in Syria, this system implements the whole "Mofadaleh" procedures, starting from adding students applications and ending with the election results.

Moreover, this system is considered as a dynamic one, as its settings allow declaring new fields by administrators. This system is used now by the Ministry and the above recommendations have been implemented to improve and measure its performance.

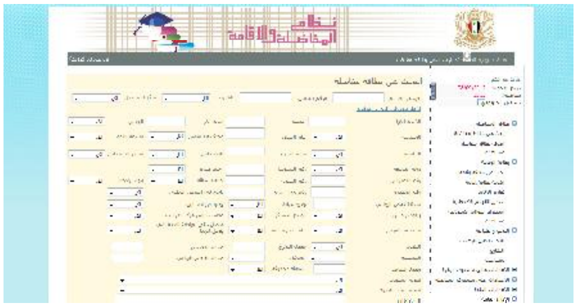

Fig. 1. "Mofadaleh" of Health Ministry in Syrian Arabic Republic

V. Applying the Recommendations on the System

The above mentioned recommendations were applied in developing the case study system and the study shows the following.

Fig. 2. Before applying any recommendations

Fig. 2 shows the files of the called main page with the size of each and the time needed to load it.

It is noticed that the number of JavaScript files is 10 and the CSS files is 8 while the number of HTTP requests is 54 requests. Additionally, the whole size of the webpage is 1.2 MB and the time needed to load it is 15.4 seconds. Fig. 3 shows the enhancements got after removing whitespaces in JavaScript and CSS files, abbreviating the names of used variables and methods and deleting all comments that had been written during the development phase.

Fig. 3. Enhancements on external objects

As noticed the size of the webpage is reduced by 0.2MB to be just 1.0MB and the time needed to load it became 14.9 seconds instead of 15.4.

Fig. 4. After applying the recommendations

This page could be uploaded by one request from the server if we merge JavaScript in one file and CSS in one file.

Fig. 4 shows that the number of needed HTTP requests is reduced to 38 requests and the time needed to load the webpage has became 9.05 and those results were achieved after merging JavaScript and CSS files into one file for each and avoiding the use of AJAX.

Fig. 5 shows performance improvement of the "Mofadaleh" system after the whole 4 most important recommendations were considered with the impact of each recommendation.

It shows that reducing the size of the header of web pages resulted the most impact with 48% performance gain, while rewriting the code resulted the least impact with 10% performance gain.

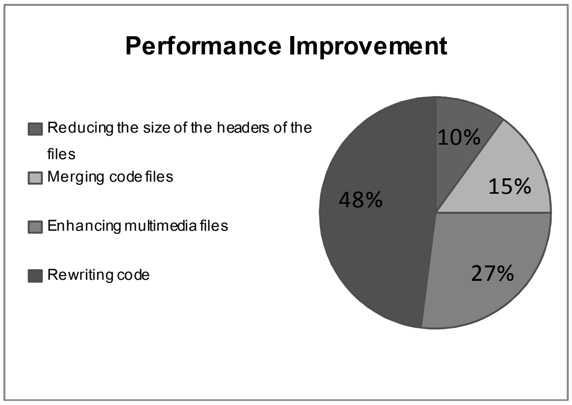

Fig. 5. Improvement in the performance according to 4 of the most important listed recommendations

VI. Conclusion

In Conclusion, this paper listed the problems that face web-based applications and several recommendations that should be considered during and after the development process of such applications. It also showed the impact of considering these recommendations after applying them in a real case which is the "Mofadaleh" system for the Syrian Health Ministry.

References

[1] Anna Bouch, Quality is in the Eye of the Beholder: Meeting Users' Requirements for Internet Quality of Service, *CHI 2008, The Hague, the Netherlands,* April 2008, pp. 297 – 304.

[2] Akamai, Retail Web Site Performance: Consumer Reaction to a Poor Online Shopping Experience (Akamai Technologies, 2006)

[3] Kenneth G. Linden, *Marissa Mayer at Web 3.0*, November 2011.

[4] Dan Farber, *Google's Marissa Mayer: Speed Wins*, November 2011.

[5] Ron Kohavi, Roger Longbotham, Online Experiments: Lessons Learned, *Computer, vol. 40 n. 9, September* 2007, pp. 103 – 105

[6] Y. Skadberg, J. Kimmel, Visitors' flow experience while browsing a Web site, *Computers in Human Behavior, vol. 20, n. 3*, May 2009, pp. 403 – 422

[7] N. Tractinsky, Evaluating the consistency of immediate aesthetic perceptions of web pages, *International Journal of Human-Computer Studies, vol. 64, n. 11,* August 2006, pp. 1071 – 1083.

[8] J. Domenech, A user-focused evaluation of web perfecting algorithms, *Journal Computer Communications, vol. 30, n. 10,* July 2007, pp. 2213 – 2224.

[9] D. Flinn, B. Betcher, *Email to author*, (Gomez, Inc., 2008).

[10] J. Yuan, A More Precise Model for Web Retrieval, *WWW 2005*, May 2005, pp. 926 – 927.

[11] H. Krichene, A. C. Ammari, A. Jemai, M. Abid, Performance/Complexity Analysis of a H264 Video Encoder, (2007) *International Review on Computers and Software (IRECOS),* 2 (4), pp. 401 – 414.

[12] S. Gulliver, G. Ghinea, Defining User Perception of Distributed Multimedia Quality, *ACM Transactions on Multimedia Computing, Communications and Applications,* 2006, pp. 241 – 257

[13] H. Zimmerman, OSI Reference Model—the IS0 Model of Architecture for Open Systems Interconnection, *IEEE Transactions on Communication, vol. COM-28, no. 4, April 1980.*

Authors' information

[1]Syrian Virtual University. Faculty of Information Technology Engineering- Damascus University, Syrian Arab Republic.

[2]Syrian Virtual University, Syrian Arab Republic.

Bassel AlKhatib is the web science master director at the Syrian Virtual University and the head of Artificial Intelligence department at the Faculty of Information Technology Engineering- Damascus University. He holds PhD degree in computer science from the University of Bordeaux-France, 1993. Dr. Alkhatib supervises many PhD students in web mining, and knowledge management. He also leads and teaches modules at both BSc and MSc levels in computer science and web engineering in both Syrian Virtual University and Damascus University.

Mouhamad Kawas holds a Master of Web Sciences from Syrian Virtual University, BS in Software Engineering and Information Systems from Damascus University.

He is an assistant teacher at the Syrian Virtual University. He achieved many awards and honors from various areas in technologies and web sciences.