PERFORMANCE THRESHOLD OF THE INTERACTIVE RASTER MAP PRESENTATION – AS ILLUSTRATED WITH THE EXAMPLE OF THE JQUERY JAVA SCRIPT COMPONENT

Karol Król, Ph.D.
Department of Land Management and Landscape Architecture
University of Agriculture in Krakow
Krakow, Poland
e-mail: k.krol@onet.com.pl – contact person

Abstract

The purpose of the present work was to determine at what size of the raster file the effective performance threshold of the map application will be exceeded, which will cause difficulties in browsing the map, resulting from, for instance, a prolonged loading time of the screen in the browser window, or from the less-than-smooth running of the application. Five model versions of the web application, differing in the size of the raster maps, were tested for performance. Applications were created using the Mapbox component: Zoomable jQuery Map. The performance of the application composed of all the raster maps, superimposed one upon another, has been assumed as the benchmark for measurements. The soil and agricultural map, created using the QGIS software, served as the raster base. Performance tests were conducted in an informal manner, using selected web applications. The values of performance indicators – Yslow and PageSpeed Score – were analysed, along with the loading time of the application in the browser window. In conclusion, it was shown that the loading time of the application in the browser window must not be equated with its general performance.

Key words: web application performance, performance tests, web mapping, raster maps, mashup

Introduction

In recent years, the number of generally available components expanding the functionality of websites – the so-called "plug-ins" – has significantly increased. The main advantage is their modularity – they can be used in many projects at the same time, as well as "switched" on and off at the user's request. Among the wide range of various components extending the functionality of websites, there are those that are responsible for the presentation of spatial data. Interactive cartographic publications made available online have many advantages. Maps are a clear and intuitive way of data presentation – whereas a combination of thematic maps often allows you to see and to show the hidden relationships in numbers. Furthermore, information provided in the form of an interactive map is attractive to the recipient, who finds it easier to remember (BROVELLI et al., 2015).

Online maps are published using various techniques and designing tools. They may be made available in the form of dynamic web applications (independent map services, geoportals) or components of other, hybrid websites (mashups). The criteria that can be decisive when selecting a given component typically include its functionality and efficient performance, often resulting from the working principle of the component itself (DANIEL et al., 2011).

Websites and web applications have increased their volume over tenfold during the last decade (ZHU, REDDI, 2013). Extensive, multimedia-based, functional websites are the result of technological advances, coupled with the expectations of the users, who are not interested in the infrastructure through which the content is delivered, only in the efficient performance with which that content can be viewed, modified, and downloaded (DICKINGER, STANGL, 2013). This generates a demand for high-performance computer systems and mobile devices that will ensure the comfort of browsing the websites, which provide increasingly complex functionalities. It also requires new design solutions that will ensure high performance (KRÓL et al., 2016).

Browsers of raster maps are intended for the publication of "ad-hoc" maps by municipalities and public administration, when there is no economic justification for creating extensive map services, and there is a need to quickly publish the map, for instance, in the case of changing the ordinal numbering of buildings (KRÓL, 2016). These solutions are usually based on raster maps. Limitations to the usability of the presentations thus made available result mainly from the size of the screen that often loads in the browser window in its entirety, which may be inefficient. The purpose of the present work was to determine at what size of the raster file the efficient performance threshold would be exceeded, causing difficulties in browsing the map.

Online maps as website components

Online maps being a component of a website (extending its functionality) are most often elicited in a browser window via a “floating frame” (iframe) or implemented using the API programming interface (BOWIE et al., 2014, PETERSON, 2015). The emergence of application programming interfaces (APIs), i.e. sets of procedures, protocols and tools for creating web applications, has contributed to the creation of numerous and increasingly accomplished mashup and hybrid applications (WOOD et al., 2007, YU et al., 2008). They combine selected thematic content with master maps from geodata providers. APIs facilitate programming of mapping services, which may include a website component, among others (SMITH, 2016). Map components can also be created using the public jQuery JavaScript library (KRÓL, SZOMOROVA, 2015). The advantage of such solutions is the small volume of the application itself, whereas its disadvantage is the dependence of the application’s performance (usability) on the size of the raster, when the component’s operation is based on its presentation. The size of the raster can therefore determine the effective performance of the component, and translate into the performance (usability) of the entire site.

Performance matters

Effectiveness of websites is considered in many aspects, but most often those relating to technology, and to sales, and it is expressed by numerous parameters. One of those parameters is performance, often equated with the speed of loading the site in the browser window. The website’s performance is largely due to the design solutions adopted, including the techniques and components that had been used to create it (WU, WANG, 2010). Currently, one of the shortcomings of websites (responsive, multimedia-based, interactive, and created on the basis of extensive content management systems – CMS) is their efficiency. Optimization of the latter can be crucial for user comfort. From the user’s point of view, performance is a measure of usability. It is a parameter that determines the comfort of browsing the website, which may have a bearing on conversion goals (SANDERS, GALLOWAY, 2013).

Website performance is considered from the perspective of server performance on which it is maintained (server-side performance, where data server is subjected to performance tests) and from the perspective of the website’s performance (client-side performance). Websites are subjected to “on input” performance tests (total load time in the browser window, and the time of reading the content shown on the display) or in the “continuous monitoring” model (when using the application under variable load, simulated or in natural conditions, i.e. during regular use).

One of the most important performance parameters is the perceived time of loading the site in the browser window – considering that around 39% of Internet users claim that the speed of the website is more important than its functionality (AKAMAI, 2017). Website performance has a significant impact on its effectiveness. Research has shown that delaying your website’s loading time in a browser window by 100 milliseconds (0.1 seconds) can lower your conversion rate by 7%. A website rendered within 10 seconds gains 46% fewer page views, and its bounce rate is 135% higher. About 53% of mobile site visitors leave it if that site loads for more than three seconds (CROSMAN, 2010). Website performance is also perceived through the lens of the Internet-specific “sense of time”. Studies have shown that, for an average user, the perceived waiting time for a website appears about 15% longer than it actually is. According to research by Google, fast-loading websites generate lower maintenance costs, they are more readily viewed, and visitors spend more time there, while even a half-second delay affects visitor statistics (SINGHAL, CUTTS, 2010).

At all stages of creating a website, but also at any time after its publication, it is useful to apply tools that automatically measure selected aspects of its technical and functional quality. The results of automated tests are usually presented in a descriptive form and synthetic point scores – expressed in the letters of alphabet or graphic symbols. A synthetic final score facilitates making a comparison between many similar projects. It also allows you to compare the measurement results obtained using various tools. This fits in with the concept of cross-measurement, consisting in testing selected parameters of the website using at least two different testing tools. Selected testing applications also present lists of critical points that need to be optimized, along with a list of post-testing recommendations.

Material and methods

Five model, twin versions of the web application were subjected to performance tests of the “client-side performance” type (Fig. 1). Each of them was created out of two raster maps (in PNG format) – the master map, identical in each version of the application, presented by default after eliciting the application in the browser window; and a base map, intended for exploration, i.e. elicited at the user’s request (Table

1). The performance of the application created from all five base maps superimposed one on top of the other (Fig. 2) was assumed as the benchmark for measurements. The soil and agricultural map of the Proszowice Municipality created using the QGIS software served as the raster basis. Proszowice Municipality is located in the northern part of the Małopolska Province (Poland) in Płaskowyż Proszowicki (the Proszowice Plateau).

Table1. Description of selected attributes of raster maps in the form of web applications.

No.	Raster size grade (dpi)	Size of the raster in pixels	Size of the raster (MB)	Total size of the application (MB)
1	72	841x595	0.33	0.66
2	150	1753x1240	0.89	1.19
3	300	3507x2480	1.99	2.32
4	450	5261x3720	3.16	3.49
5	600	7015x4960	4.34	4.67
6	Reference application	from 841x595 to 7015x4960*	10.7	10.8

*Key: *raster maps elicited incrementally as a result of user activity.*

Source: Own study.

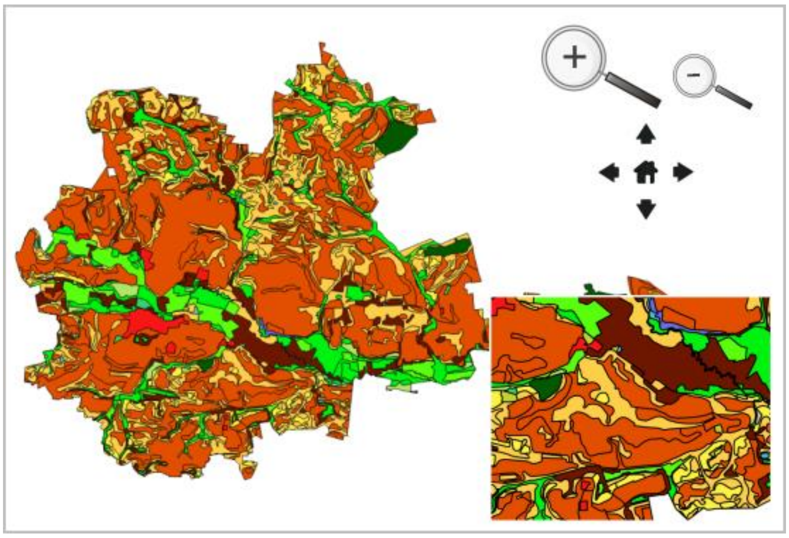

Fig. 1. View of the application allowing the browsing of a raster map.
Source: Own study using the Mapbox component .

Performance tests were performed in an informal manner, using selected online applications (Table 2). The tests were carried out for both mobile devices (in mobile mode) and desktop computers (in desktop mode).

Table 2. Online applications used in performance testing.

Testing application	Measurement unit
PageSpeed Insights (Google Developers)	PSI D, PSI M attributes
GTmetrix	PageSpeed Score, YSlow
Pingdom Website Speed Test	Google PageSpeed Performance grade, load time, page size
Dareboost: Website Speed Test (DaM)	Performance grade, Speed Index
Website Speed Test – Image Analysis Results	Page Image Score, Total Image Weight (TIW), Potential Compressed Weight (PCW)

Source: Own study.

Page Speed Insights measures site performance on mobile devices (PSI M) and desktop computers (PSI D). The application downloads the resources available at the URL twice – through a mobile client, and through a desktop client – it then measures the time of loading the part of the page visible on the screen, and the time in which the page loads fully. PageSpeed Insights checks whether a site can be generated on a mobile device in less than one second. In the "mobile" test, the time of generating a fragment of the site visible on the screen (Above The Fold – ATF) is significant. The result of the measurement is a score between 0 and 100 percentage points (PageSpeed Score). A rating of at least 85 points means that the page performance is relatively good, but its selected parameters can be optimized.

The GTmetrix application measures the website's performance, its loading time in the browser window, and the sizes of its components. The result of the performance measurement is presented using the PageSpeed Score and YSlow indices. The YSlow index is an alternative to the Google PageSpeed Score, created and shared by Yahoo! The YSlow attribute is expressed with a synthetic point score, ranging between 0 and 100 percentage points.

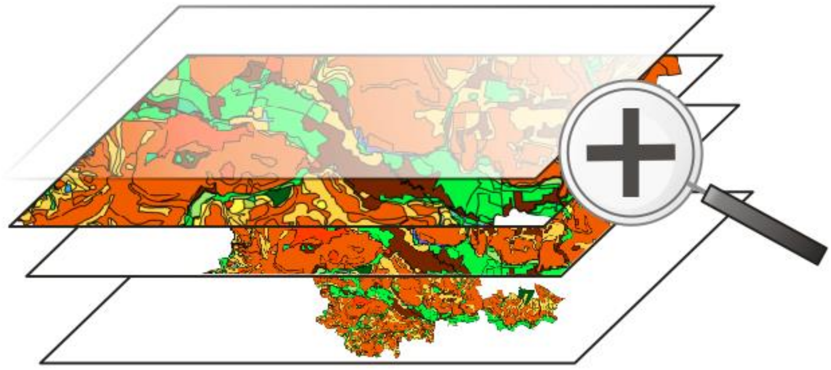

Fig. 2. Schematic arrangement of raster maps, superimposed one upon another, and subsequently elicited using Mapbox.
Source: Own study.

The Pingdom Website Speed Test application, like GTmetrix, provides information about the website's performance, its loading time in the browser window, and the sizes of its components. Dareboost (DaM) provides a module for testing the applications in the mobile mode. The result of the measurements is presented in the form of a synthetic point score and the Mobile Speed Index. The faster the rendering (that is, the better the performance), the lower the index value, with Google recommending that it does not exceed 1000 units. The Website Speed Test Image Analysis Tool application, on the other hand, analyses the graphic files that make up the website being tested. The result of the "degree of image compression" test is expressed by a synthetic indicator, the Page Image Score. The application's algorithm identifies graphic files that make up the site, measures their volumes, and provides information about the possibilities of their compression.

Model applications were created using HTML, CSS and the Mapbox component: Zoomable jQuery Map Plugin (MIT License) (MOHLER, 2018). The jQuery Mapbox Plugin facilitates the presentation of raster graphics. The applications served as raster browsers (Image Viewer) and were developed to consume minimal server resources, as well as to not overburden the Internet connection. The applications were extended by adding a graphical user interface (GUI), which facilitates viewing the map using the navigation icons.

Among other things, Mapbox allows horizontal viewing of raster graphics (grab and drag the map area). Also, it simulates the effect of zooming the map view, by overlapping appropriately prearranged vertical rasters. Zooming in the map view consists in the presentation of previously developed rasters. The rasters are entered into the structure of the hypertext document, and displayed in the browser window one under the other. As a result of user activity, individual variations of the raster are elicited, which constitutes a kind of simulation of zooming in on the map view. Mapbox component: Zoomable jQuery Map Plugin was chosen due to its low hardware requirements and lack of restrictions as to the size of the presented raster graphics.

Outcome of the study

The performance of individual versions of the application on desktop devices gradually decreased with the increase in the size of the master map raster (Table 3), which was correlated with the loading time of the application's components (Fig. 3). An increase in the raster size caused difficulties in browsing the map. With the raster size equal to 150dpi, "step loading" of individual rasters was noticeable, and with the size exceeding 300dpi, the application lost its smooth operation. Thus, the measurements demonstrated that with a 150dpi grid, the effective performance threshold, at which relatively comfortable use of the application is possible, had been exceeded.

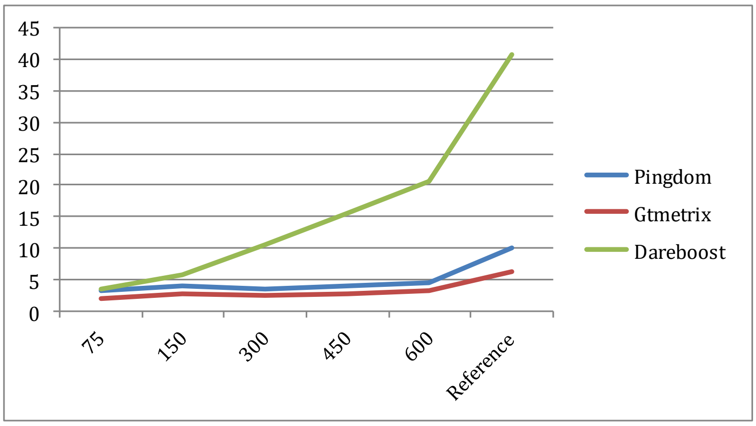

Fig. 3. Loading time of the application in the browser window depending on the raster size.
Source: Own study.

The performance results obtained in the PSI M test are puzzling. They suggest that the performance of applications on mobile devices improves with the increase of the raster size. It should be noted here that the PSI M performance measurement is carried out without full rendering, i.e. not all of the resources constituting the tested application are loaded during the process. However, this does not explain the increase in the PSI M performance index, despite the increase in the raster size. In turn, the results obtained using the Dareboost application show a drop in the application performance on mobile devices with the increase of the raster size. Admittedly, these are not big drops. Everything seems to indicate that application architecture (i.e. the way it was created) is significant for performance. Performance is measured only for those components that are visible to the user when loading the application in the browser window. Other resources, in this case those with the most impact on performance, are elicited when using the application. Then, there may also be a decrease in the application's performance (that will not be seen in the state of "application rest"). This is indicated by the value of the Dareboost Mobile Speed Index. Its value remains at a similar level regardless of the size of the base raster (Table 3).

The tests showed that the size of the YSlow index does not depend on the size of the application's components that are "waiting to be elicited" (measurement without full rendering). The application was created in such a way that the base map raster (72dpi) would be read in the browser window first, while the master map raster (150-600dpi) would remain "available" to the user, that is, eliciting it in the browser window would occur only as a result of user activity. The base for each version of the tested application was therefore the same; it was the master map rasters that were different. This may translate to the identical size of the YSlow parameter; also when measuring the performance using the Pingdom Website Speed Test application.

Table 3. Measurement results for selected performance indices.

Testing application	Reference application		Raster size (dpi) 72		150		300		450		600	
PSI	M	D	M	D	M	D	M	D	M	D	M	D
	83	23	67	83	70	74	75	59	75	51	77	43
GTmetrix	P	Y	P	Y	P	Y	P	Y	P	Y	P	Y
	24	77	83	77	74	77	59	77	51	77	45	77
Pingdom	88		88		88		88		88		88	
Dareboost Mobile	58		61		58		54		54		55	
Dareboost Mobile Speed Index	4297		2419		2379		2410		2389		2410	

Key: M – mobile; D – desktop; P – PageSpeed Score; Y – YSlow.

Source: Own study.

The loading time of the developed applications is not satisfactory. The application presenting the 72dpi raster loaded in the browser window in just over 3 seconds (according to Pingdom and Dareboost), while the loading of the 600dpi map took up to 7 times longer on mobile devices (Table 4). The loading time of an application presenting 300dpi and larger rasters exceeded 10 seconds. It was therefore below users' expectations. Such presentation does not meet performance standards. Furthermore, the measurements showed that the loading time of the application must not be equated with its general performance (efficiency). The latter remained at a similar level, while the loading time of parts of the application extended with the increase in the size of the raster.

Table 4. Loading time of the application in the browser window depending on the size of the raster map.

Testing application	Reference application		Raster size (dpi) 72		150		300		450		600	
	L	S	L	S	L	S	L	S	L	S	L	S
Pingdom Desktop	10.01	10.8	3.32	0.55	3.88	1.1	3.62	2.2	4.09	3.4	4.57	4.6
GTmetrix Desktop	6.2	10.8	1.9	0.55	2.8	1.1	2.5	2.2	2.8	3.38	3.2	4.56
Dareboost Mobile	40.86	11.07	3.42	0.56	5.72	1.13	10.66	2.32	15.56	3.54	20.54	4.78

Key: L – loading time, S – total page size.

Source: Own study.

The Website Speed Test application has a programmed limit (protection against excessive server load), which users are not informed about. The measurement results were only available for the application presenting the master map raster in a size not exceeding 300dpi (Table 5). This was demonstrated by the reference application test, which consisted of five rasters in the size from 841x595 (72dpi) to 7015x4960px (600dpi). In the test results, it was noted that only those rasters whose size did not exceed 300dpi had been verified. The remaining rasters had been omitted. In the case of tests that were performed in full, in each case the application indicated the possibility of raster compression by about 30%.

Table 5. Website Speed Test – Image Analysis Results.

Index	72		150		300		450; 600; Reference	
Page Image Score (PIS)	B (good)		B (good)		B (good)		no data	
Image Weight Comparison (IWC)	TIW	PCW	TIW	PCW	TIW	PCW	TIW	PCW
	0.507	0.157/30.9	1.1	0.3/27.8	2.3	0.616/27.2	no data	no data

Key: TIW – Total Image Weight (MB), PCW – Potential Compressed Weight (MB/%).

Source: Own study.

Conclusions

The tests have shown that the division into raster graphics (maps) prepared for online publication (72dpi) and those intended for printed publications (from 300dpi upwards) is justified. The dynamics of loading raster graphics decreases along with their size. The effective performance threshold, also understood as the threshold of map viewing comfort, or the flexibility threshold, depends on the device on which the map is displayed. The application's architecture should be different for mobile devices than for desktop devices. Therefore, the way in which the map presentation will be programmed is of great importance. An example of the image viewer that was developed for the presentation of large raster files is Zoomify Viewer. Presentation of the raster in the application window, however, follows the division of the raster into smaller fragments, which are then elicited one by one. This significantly increases the flexibility of the application and determines its usability. The dynamics of the application may also be dependent on the software, including the system platform and the web browser operated by the user.

When there is a need to publish a large raster in its entirety – the size of which reduces the efficiency of the application – it should be equipped with graphics presenting the progress of loading the raster map, the so-called "progress bar". It will focus the user's attention and reduce the sense of discomfort resulting from long waiting.

It is difficult to set one generally accepted performance threshold, characteristic for all web applications presenting raster maps, at which it is displayed in a dynamic (i.e. efficient) way. These applications are executed in various ways, using various programming tools and techniques. Therefore, it is possible to influence application performance not only by reducing the size of the raster (compression), although it may translate into a reduction in its adaptability, but also by changing the technique of its presentation (programming changes). Improving the performance of the application requires multiple measurements performed according to the A/B test scenario. This will lead to rejecting design concepts that would reduce application performance. In addition, automated (algorithmic) tests can be supplemented by a performance assessment by the users.

Performance tests demonstrated that the first time the application is loaded in the browser window, it runs relatively smoothly. Therefore, the measurement of application performance should be carried out during its use (in monitoring mode), using various testing tools, because it is then that the most problems with its performance may occur – during raster elicitation and browsing or during “raster switching” during which the rasters are elicited by the user when trying to zoom in on the map view. Therefore, the biggest fluctuations in performance may occur during the use of the application, and not during the first loading of the application in the browser window. Delays and lack of smoothness in the raster presentation during its browsing may be decisive for the user in terms of discontinuing to use the application. It is therefore important to equip the application with mechanisms that provide feedback in the form of graphical information about the loading progress (status) of the application.

References

AKAMAI, 2017. *Akamai Online Retail Performance Report: Milliseconds Are Critical*. State of Online Retail Performance report, https://goo.gl/72fpyT (access: 24.01.2018).

BOWIE, G.D., MILLWARD, A.A., BHAGAT, N.N. 2014. *Interactive mapping of urban tree benefits using Google Fusion Tables and API technologies*. Urban Forestry & Urban Greening, 13(4): 742-755.

BROVELLI, M.A., ZAMBONI, G., MUNOZ, C. A. 2015. *From paper maps to the Digital Earth and the Internet of Places*. Rendiconti Lincei, 26(1): 97-103.

CROSMAN, P. 2010. *Online Banking Customers Expect Fast Website Performance, Survey Finds*. InformationWeek: Connecting The Business Technology Community, https://goo.gl/FdWDyH (access: 24.01.2018)

DANIEL, F., MATERA, M., WEISS, M. 2011. *Next in mashup development: User-created apps on the web*. IT Professional, 13(5): 22-29.

DICKINGER, A., STANGL, B. 2013. *Website performance and behavioral consequences: A formative measurement approach*. Journal of Business Research, 66(6): 771-777.

KRÓL, K. 2016. *Data presentation on the map in Google Charts and jQuery JavaScript technologies*. Geomatics, Landmanagement and Landscape, 2: 91-106.

KRÓL, K., SZOMOROVA, L. 2015. T*he possibilities of using chosen jQuery JavaScript components in creating interactive maps*. Geomatics, Landmanagement and Landscape, 2: 45-54.

KRÓL, K., SZEWCZYK, B., PAWŁOWSKA, B. 2016. *Interaktywna prezentacja zagadnień środowiskowych za pomocą Google Fusion Tables na przykładzie zdjęć fitosocjologicznych wybranych okolic Tenczynka (Interactive presentation of environmental issues using Google Fusion Tables on the example of phytosociological images of selected Tenczynek areas)*. Acta Sci. Pol., Formatio Circumiectus, 15(4): 253-264.

MOHLER, A. 2018. *Mapbox: Zoomable jQuery Map Plugin*, https://goo.gl/2WIKh (access: 20.01.2018).

PETERSON, M.P. 2015. *Evaluating mapping APIs*. In: J. Brus, A. Vondrakova, V. Vozenilek (Editors). Modern Trends in Cartography. Springer, Cham., 183-197.

SANDERS, J., GALLOWAY, L. 2013. *Rural small firms' website quality in transition and market economies*. Journal of Small Business and Enterprise Development, no. 20(4): 788-806.

SINGHAL A., CUTTS, M. 2010. *Using site speed in web search ranking. Google Webmaster Central Blog*, https://goo.gl/sVusvY (access: 24.05.2017).

SMITH, D. A. 2016. *Online interactive thematic mapping: Applications and techniques for socio-economic research*. Computers, Environment and Urban Systems, 57: 106-117.

WOOD, J., DYKES, J., SLINGSBY, A., CLARKE, K. 2007. *Interactive visual exploration of a large spatiotemporal dataset: reflections on a geovisualization mashup*. IEEE Transactions on Visualization and Computer Graphics, 13(6): 1176-1183.

WU, Q., WANG, Y. 2010. *Performance testing and optimization of J2EE-based web applications*, In: Second International Workshop on Education Technology and Computer Science, ETCS, 2: 681–683.

YU, J., BENATALLAH, B., CASATI, F., DANIEL, F., 2008. *Understanding mashup development*. Internet Computing, IEEE Internet Computing, 12(5): 44-52.

ZHU, Y., REDDI, V.J. 2013. *High-performance and energy-efficient mobile web browsing on big/little systems*. In International Symposium on High Performance Computer Architecture (HPCA), 13–24.