AN ANALYSIS OF TECHNIQUES AND QUALITY ASSESSMENT FOR WEB PERFORMANCE OPTIMIZATION

Shailesh K S
Research Scholar,SOCIS, IGNOU, Maidan Garhi, Delhi - 110068
Shailesh.shivakumar@gmail.com

Dr. P. V. Suresh
Director, SOCIS, IGNOU, Maidan Garhi, Delhi - 110068
pvsuresh@ignou.ac.in

Abstract - **Performance of web applications impact the user satisfaction, online revenue, search engine ranking, site traffic, SEO and other such key metrics. In this paper we have conducted the survey of state of the art tools, techniques, methodologies of various aspects of web performance optimization. We have elaborated key aspects of pro-active quality in web applications. We have elaborated on various techniques related to WPO such as asset optimization, monitoring and testing.**

1. INTRODUCTION TO WEB PERFORMANCE OPTIMIZATION (WPO)

Web performance impacts the revenue [2] and ranking of the web pages [1], It also impacts the online revenue [56] [57]. Site performance also impacts the success rate [71].

WPO involves best practices and techniques to increase the speed of web pages [87]. WPO includes page components such as HTML content, presentation components, page elements, page assets and such. WPO provides techniques, best practices, thumb rules, methodologies for end-to-end web performance optimization.

1.1 Impact of WPO

WPO has impact in following aspects:

- Customer churn: Research indicates that customers would abandon the slower web pages [88] [91] [92]
- User impact: User experience is drastically impacted due to page performance
- Site Traffic: Site traffic is impacted if the page takes more than 3 seconds [89] and most users expect the page to load within 2 seconds [89].
- Revenue: For e-commerce sites, the shopping and check out performance is crucial for the conversion
- Multi-device optimization: An optimized web page also impacts the performance on various devices

1.2 Dimensions of WPO

The process of Web optimization can be analyzed from several dimensions:

- **Optimization of Request pipeline processing Systems:** In this category we will examine all the systems involved in the web request processing pipeline. This involves browser software, CDN, proxy server, network, load balancer, web server, application server, integration middleware, backend services, database server and such
- **Client-side and server side optimization**: Client-side performance optimization includes all optimizations performed on client-side presentation components such as HTML pages, images, assets and such. Server side optimization includes performance tuning of server –side components such as fine-tuning business components, setting optimal server configuration, right infrastructure sizing and such
- **Design time and run time optimization:** Design time optimization includes the static and offline performance optimizations activities such as performance code reviews, performance testing, and offline performance tuning and such. Run time and dynamic performance optimization activities include real-time performance monitoring and notification, run time performance optimization and such.
- **Web component optimization:** Another aspect of web optimization is to optimize each of the constituent's web components such as HTML, images, JavaScript, CSS, Rich media files and such.

2. THE ROLE OF SECURITY IN WEB PERFORMANCE OPTIMIZATION

In this section we briefly discuss the impact of security on performance.

2.1 Security & WPO

Security is one of the key concerns for web applications. Security for web applications can be enforced at various levels. One of the most commonly used security constraint is to use secured socket layer (SSL) to ensure transport level security. SSL is a default choice for web pages hosting confidential information such as user credentials, user personal information and such. SSL would also impact the page performance [78] due to additional overhead. The most commonly used techniques for optimizing performance in such scenarios are as follows:

- Set proper expiration times for the objects so that browser can cache the objects appropriately [74]
- Use CDN which support SSL acceleration modules for forward caching

3. WEB PERFORMANCE DESIGN CHECKLIST

In this section, we will look at the design guidelines and best practices for achieving optimal web performance. The books [66] [67] provide excellent performance guidelines from web performance stand point.

Table 1: Categorized Performance Rules

Category	Performance Rule	Impact on web performance
Request Optimization	Reduce the number of HTTP Requests	Reduces the consumed bandwidth and data transferred o
	Merge the static assets such as JS and CSS files	Merging would reduce the number of HTTP requests and would improve the page response times by about 38% [66]
	Remove all duplicate file includes	
	Remove all invalid URLs which result in HTTP 404	Avoids unnecessary and invalid HTTP requests
	Load the JavaScriptsascynhronously	This would reduce the blocked loading of JS files
	Minimize usage of iframes	iFrames block the loading of parent window till iframe source is loaded and hence affects load times
	Minimize redirects	Minimize additional requests
	Cache DNS records	We can reduce the DNS lookup time through DNS cache maintained at browser level
	Remove any unused CSS, JS file includes	Minimizes HTTP requests
Web object size optimization	Minify JS and CSS files	Minification would reduce the size of JS and CSS file
	Compress images	Compresed images would reduce overall page size
	Leverage gzip for HTTP compression	Compression would reduce the overall page size by about 70% [66]
	Remove white space in the HTML document	Removal of white space would reduce the overall page size
HTTP Header Optimization	Leverage cache headers for static assets (images, JS, CSS, JSON and other binary files) using Cache-Control header with max-age directive	Allows browsers to optimally cache the assets
	Use expires header for the assets	Avoids additional resource request
Asset placement	Place CSS files at the top	CSS elements in the head tag
	Place JS files at the bottom	JS files at the bottom would improve the perceived page load time. I would

		avoid the blocked loading of other assets
	Externalize inline JS or CSS	Enables browser caching and parallel downloads
Image optimization	Asynchronous image load	Load the images on-demand and in asynchronous when they are visible in the user's view port.
	Optimize image size	Use the right size image based on the requesting device
	Convert JPEG image formats to progressive format	This would reduce the overall image size
	Optimize image dimensions	Specify the exact width and height for all images
	Use image maps	Reduces multiple image requests
	Use CSS sprites	All images are combined into a single one and the required image is displayed using style rules. This reduces number of image requests
	Other techniques: inline images	
Network optimization	Usage of CDN	CDN would optimize the resource request by serving the resource from nearest location to the requestor
	Use multiple asset hosting servers (for hosting images, videos and other multi-media content)	Allows browsers to download the content in parallel.
External Dependency optimization	Identify all external scripts and HTTP requests which impact the page performance and which block the page load and optimize them	
Web Application design optimization	• Perform regular and iterative performance testing to identify performance bottlenecks and fix them. Use automated and manual performance code reviews at regular intervals. • Use light-weight service based integration model and load the data asynchronously on-demand	Iterative performance testing uncovers performance bottleneck during early stages

3.1 Tools for web performance optimization

Table 3. below provides a list of tools that can be used for web performance optimization

Testing Type	Open Source Tool(s)
Web testing	• Selenium (http://www.seleniumhq.org/) • Crawljax • Cucumber • HtmlUnit
Accessibility Testing	• W3c Accessibility validator (http://www.w3.org/standards/webdesign/accessibility)
HTML Compliance Testing	• W3c validator (http://validator.w3.org/)
Security Testing	• OWASP.org web security testing tools

Web Performance Analysis	- Google PageSpeed (https://developers.google.com/speed/pagespeed/) - Yahoo Yslow (http://developer.yahoo.com/yslow/)
Performance Testing	- Jmeter (http://jmeter.apache.org/) - LoadUI
Static Code Analysis Tools	- Checkstyle (http://checkstyle.sourceforge.net/) - PMD (http://pmd.sourceforge.net/) - Findbugs (http://findbugs.sourceforge.net/) - SonarSource (http://www.sonarsource.com/)
Continuous Integration Tools	- Jenkins (http://jenkins-ci.org/) - Hudson (http://hudson-ci.org/)
Profiling Tool	- Jprofiler
Service Testing	- SOAPUI (http://www.soapui.org/)

4. ASSET OPTIMIZATION

Static assets such as images, multi-media files and other binary files contribute to the web performance. On an average 70-80% of page contains non-HTML content such as images, JS, CSS [58] [66]. Hence optimizing the digital assets would be crucial for optimizing the performance of the overall web page.The books [66] [67] provides following techniques for optimizing the asset performance

Page assets apart from the page HTML include the image graphics, media files (videos, flash), JavaScript files, Stylesheet files (CSS files) and JSON based data.

Let us look at optimizing each of these assets in table 4:

Table 2: Asset Optimization Techniques

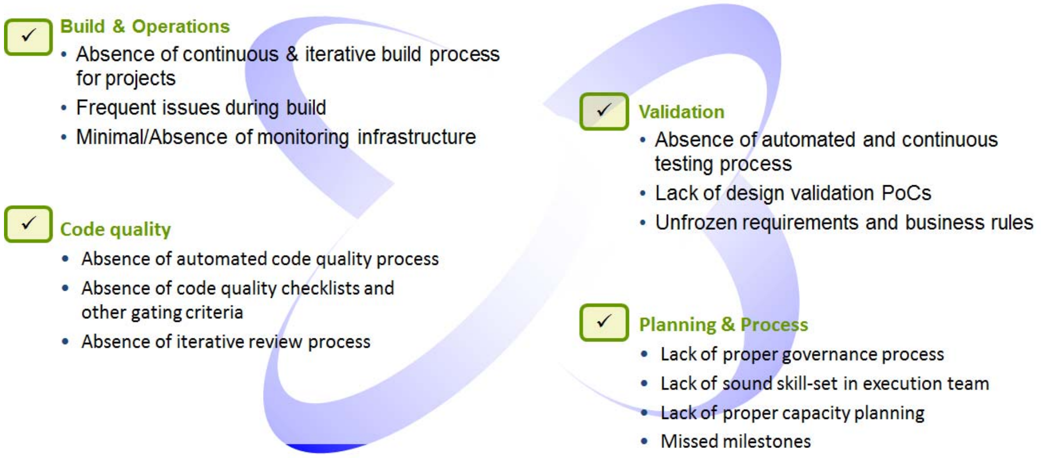
Asset Category	Optimization Techniques
Images	- Use right format image for the appropriate channel. - Use CSS sprites to minimize image load requests - Compress images for optimal image sizing - We can resize the image with minimal size in PNG format - Use adaptive images to cater to multiple devices
Resource requests	Minimize resource requests by merging CSS/JS files to reduce multiple file requests
JS/CSS Optimization	All the JavaScripts and StyleSheets should be merged and minified. This would not only reduce the HTTP requests but also reduce the impact on overall page size. Also we can place the merged and minified file at the bottom of the page to optimize the perceived page load times
Asynchronous asset loading	Wherever possible, load the assets such as images asynchronously and only on demand. For instance we need not load the image that is outside of the current view until the user does a page scroll.
Web-friendly data format	Use optimal data container such as JSON instead of using XML. JSON is optimized for web and is the lightweight alternative to XML.

5. PRO-ACTIVE QUALITY IN WEB APPLICATIONS

In this section we will look at pro-active measures we can take for pro-active quality in web applications from performance stand point

5.1 Common Lead Indicators of a Future Quality Problems

Lead Indicators act as early indicators of potential quality issues which provides opportunity for project managers and project architects to pro-actively undertake course corrective actions. The key lead indicators are depicted in following diagram



✓ Build & Operations

- Absence of continuous & iterative build process for projects
- Frequent issues during build
- Minimal/Absence of monitoring infrastructure

✓ Code quality

- Absence of automated code quality process
- Absence of code quality checklists and other gating criteria
- Absence of iterative review process

✓ Validation

- Absence of automated and continuous testing process
- Lack of design validation PoCs
- Unfrozen requirements and business rules

✓ Planning & Process

- Lack of proper governance process
- Lack of sound skill-set in execution team
- Lack of proper capacity planning
- Missed milestones

5.2 Achieve Quality at Source

Given below are few methods to achieve the quality at the source.

Static Code Analyzers

- Automated IDE plugins which inspect the code for standards violation using customizable/configurable inspectors, duplicate code, critical issues such as memory/connection leaks
- Highly effective during unit testing stage
- Should be used as gating criteria prior for code check-in
- Examples include Checkstyle, PMD, Findbugs, SonarSource

Coding Checklists

- List of coding standards and best practices for programming languages
- Acts as an effective tool in self-review and peer-review
- Infosys standard checklists can be customized to suit project/client requirements

5.3 Continuous integration

Continuous Integration involves frequent and iterative process of building code, testing, integrating and deploying releases to catch issues early in the game. Various tools involved in this process are as follows:

1. **Build**
    1. Integrated code compilation – Maven
    2. Continuous build – Hudson, Jenkins
    3. Build reports - Hudson

2. Code analysis

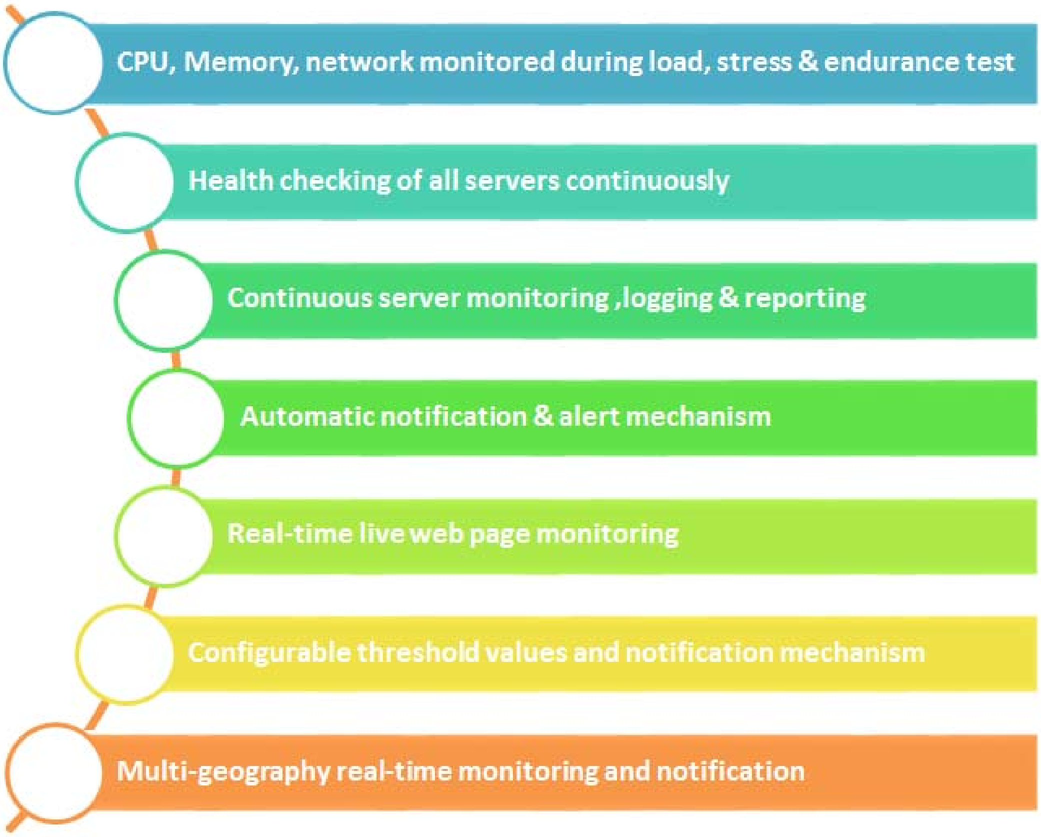
- o Duplicate code analysis – PMD
- o Coding guidelines – Findbugs, CheckStyle, Sonar

3. **Validators**
    - o Java code unit testing - Junit
    - o Web Testing – Selenium
    - o Defect report -- Bugzilla
    - o **Risk tracking**
    - o Business risk tracking and monitoring
    - o Technical risk tracking and monitoring

- Milestone reports
- **Automated jobs**
- Automated continuous deployment – Hudson, Jenkins
- Automatic notifiers – Email, log
- Automated workflows - Hudson, Jenkins
- Automated application monitoring

5.4 Internal and external monitoring for better QoS

Various aspects of internal and external monitoring is depicted in following diagram:


CPU, Memory, network monitored during load, stress & endurance test
Health checking of all servers continuously
Continuous server monitoring ,logging & reporting
Automatic notification & alert mechanism
Real-time live web page monitoring
Configurable threshold values and notification mechanism
Multi-geography real-time monitoring and notification

6. PERFORMANCE FOCUSED DEVELOPMENT & TESTING

Performance focused development and testing involves the following:

- ***Performance Testing***
  - Test performance of integrated application early in the game
  - Perform continuous performance testing
  - Open Source tools: Jmeter (for load testing), Load UI, SOAP UI (for web service testing)
- ***Automated tools***
  - Tools at various phases of the project
  - Yahoo Yslow, Google PageSpeed for measuring page performance in development
  - Profiling tools such as Jprofiler during troubleshooting phase and load testing tools during performance testing
- ***Monitoring***
  - Well-defined real-time application monitoring & notification process
  - Tools: Gomez and Web Analytics tools such as Omniture, Google Analytics

7. CONCLUSION

In this paper we examined various aspects of web performance optimization. We detailed various web performance optimization techniques such as asset optimization, caching, performance design checklist, pre-fetching, performance testing, performance monitoring.

Finally we have elaborated various methods and processes involved in pro-active quality in web applications.

REFERENCES

[1] Google. Using site speed in web search ranking. http://googlewebmastercentral.blogspot.com/2010/ 04/using-site-speed-in-web-search-ranking.html, April 9, 2010.

[2] D. F. Galletta, R. Henry, S. McCoy, and P. Polak. Web Site Delays: How Tolerant are Users? Journal of the Association for Information Systems, 5(1), 2004.

[3] Shopzilla: faster page load time = 12% revenue increase. http://www.strangeloopnetworks.com/ resources/infographics/web-performanceand-ecommerce/shopzilla-faster-pages-12-revenue-increase/

[4] Jia Wang. 1999. A survey of web caching schemes for the Internet. SIGCOMM Comput. Commun. Rev. 29, 5 (October 1999), 36-46.

[5] A. Chankhunthod, P. B. Danzig, C. Neerdaels, M. F. Schwartz, and K. J. Worrel, A hierarchical Internet object cache, Usenix'96, January 1996.

[6] Z. Wang, Cachemesh: a distributed cache system for World WideWeb, Web Cache Workshop, 1997.

[7] L. Fan, P. Cao, J. Almeida, and A. Z. Broder, Summary cache: a scalable wide-area Web cache sharing protocol, Proceedings of Sigcomm'98.

[8] S. Michel, K. Nguyen, A. Rosenstein, L. Zhang, S. Floyd and V. Jacobson, Adaptive Web caching: towards a new caching architecture, Computer Network and ISDN Systems, November 1998

[9] J. Yang, W. Wang, R. Muntz, and J. Wang, Access driven Web caching, UCLA Technical Report #990007

[10] C. R. Cunha, and C. F. B. Jaccoud, Determining WWW user's next access and its application to pre-fetching, Proceedings of ISCC'97: The second IEEE Symposium on Computers and Communications, July 1997

[11] E. Cohen, B. Krishnamurthy, and J. Rexford, Improving end-to-end performance of the Web using server volumes and proxy _fillters, Proceedings of Sigcomm'98

[12] E. P. Markatos and C. E. Chronaki, A TOP-10 approach to prefetching on Web, Proceedings of INET'98.

[13] T. Palpanas and A. Mendelzon, Web prefetching using partial match prediction, Proceedings of WCW'99.

[14] E. Cohen, B. Krishnamurthy, and J. Rexford, Efficient algorithms for predicting requests to Web servers, Proceedings of Infocom'99

[15] E. Levy-Abegnoli, A. Iyengar, J. Song, and D. Dias, Design and performance of Web server accelerator, Proceedings of Infocom'99.

[16] J. Challenger, A. Iyengar, K. Witting, C. Ferstat, and P. Reed. A Publishing System for Efficiently Creating Dynamic Web Content. In Proceedings of IEEE INFOCOM 2000, March 2000.

[17] D. C. Verma. Content Distribution Networks: An Engineering Approach. John Wiley & Sons, 2002.

[18] Arun Iyengar, Erich M. Nahum, Anees Shaikh, and RenuTewari. 2002. Enhancing Web Performance. In Proceedings of the IFIP 17th World Computer Congress - TC6 Stream on Communication Systems: The State of the Art, Lyman Chapin (Ed.). Kluwer, B.V., Deventer, The Netherlands, The Netherlands, 95-126.

[19] Killelea, P. (2002). Web Performance Tuning: speeding up the web. " O'Reilly Media, Inc.".

[20] http://httpd.apache.org/docs/current/misc/perf-tuning.html

[21] Cardellini, V., Colajanni, M., & Yu, P. S. (1998). Dynamic load balancing on scalable Web-server systems. Yorktown Heights, NY: IBM T.J. Watson Research Center.

[22] U. Acharjee, Personalized and Artificial Intelligence Web Caching and Prefetching. Master thesis, University of Ottawa,Canada(2006).

[23] Y.f. Huang and J.M. Hsu, "Mining web logs to improve hit ratios of prefetching and caching". Knowledge-Based Systems, 21(1), (2008), pp. 62- 69.

[24] G. Pallis, A. Vakali, and J.Pokorny, "A clustering-based prefetching scheme on a Web cache environment", Computers and Electrical Engineering, 34(4), (2008). pp.309-323

[25] A.K.Y. Wong, " Web Cache Replacement Policies: A Pragmatic Approach", IEEE Network magazine, 20(1), (2006), pp.28–34.

[26] H.T. Chen, Pre-fetching and Re-fetching in Web caching systems: Algorithms and Simulation, Master Thesis, TRENT UNIVESITY, Peterborough, Ontario, Canada(2008).

[27] T.Chen, "Obtaining the optimal cache document replacement policy for the caching system of an EC Website", European Journal of Operational Research.181(2),(2007), pp. 828. Amsterdam.

[28] C. Kumar and J.B. Norris, "A new approach for a proxy-level Web caching mechanism", Decision Support Systems, Elsevier, 46(1), (2008), pp.52-60.

[29] C. Kumar, "Performance evaluation for implementations of a network of proxy caches", Decision Support Systems, 46(2), (2009), pp. 492-500.

[30] J. Domenech, J. A. Gil, J. Sahuquillo, and A. Pont, "Using current web page structure to improve prefetching performance", Computer Network Journal, 54(9), (2010), 1404-1417.

[31] V. N. Padmanabhan, and J. C. Mogul , "Using predictive prefetching to improve World Wide Web latency", Computer Communication Review, 26(3), (1996). pp. 22–36.

[32] J. Cobb, and H. ElAarag, "Web proxy cache replacement scheme based on back-propagation neural network", Journal of System and Software, 81(9), (2008), pp. 1539-1558.

[33] W. Ali, and S.M. Shamsuddin, "Intelligent Client-side Web Caching Scheme Based on Least recently Used Algorithm and Neuro-Fuzzy System", The sixth International Symposium on Neural Networks(ISNN 2009), Lecture Notes in Computer Science (LNCS), Springer-Verlag Berlin Heidelberg , 5552, (2009), pp. 70–79.

[34] H. ElAarag and S. Romano, "Improvement of the neural network proxy cache replacement strategy", Proceedings of the 2009 Spring Simulation Multiconference,(SSM'09), San Diego, California, (2009), pp: 90.

[35] S. Sulaiman, S.M. Shamsuddin, F. Forkan, and A. Abraham, "Intelligent web caching using neuro computing and particle swarm optimization algorithm", Proceedings of the 2008 Second Asia International Conference on Modelling & Simulation (AMS 08), IEEE Computer Society, (2008), pp. 642-647

[36] W. Tian, B. Choi, and V.V. Phoha,"An Adaptive Web Cache Access Predictor Using Neural Network". Proceedings of the 15th international conference on Industrial and engineering applications of artificial intelligence and expert systems: developments in applied artificial intelligence, Lecture Notes In Computer Science(LNCS), Springer- Verlag London, UK 2358,(2002).450-459.

[37] E. P. Markatos and C. E. Chronaki , "A Top-10 approach to prefetching on the Web ", Proceedings of INET'98 Geneva, Switzerland, (1998), pp. 276-290.

[38] Y. Jiang, M.Y Wu, and W. Shu, “Web prefetching: Costs, benefits and performance”, Proceedings of the 11th International World Wide Web Conference, New York, ACM, (2002).

[39] N. Tang and R. Vemuri, “An artificial immune system approach to document clustering”, Proceedings of the Twentieth ACM Symposium on Applied Computing. SantaFe, New Mexico, USA , (2005) ,918-922.

[40] W. Ali, S. Shamsuddin, A. Ismail, A survey of web caching and prefetching, Int. J. Adv. Soft Comput. Appl. 3 (1) (2011) 18–44

[41] M. Liu, F. Wang, D. Zeng and L.Yang, “An Overview of world wide Web Caching”, International conference on Systems Man and Cybernetics, IEEE, 2001, pp.3045-3050.

[42] Barish, G., &Obraczka, K.. (2000). World Wide Web Caching: Trends and Techniques.

[43] Alexander P. Pons. “Improving the performance of client web object retrieval”, Journal of Systems and Software, Vol. 74, Issue 3, 2005

[44] Lei Shi; Ying-Jie Han; Xiao Guang Ding; Lin Wei. “An SPN based Integrated Model for Web Prefetching and Caching”, Springer Journal of Computer Science and Technology, Vol. 21, Issue 4, 2006.

[45] SajidHussai; Robert D.McLeod. “Intelligent Prefetching at a Proxy Server”, Proceedings IEEE Conference on Electrical and Computer Engineering, Cananda.

[46] Yi- Hung; Arbee L.P. Chen. “Prediction of Web Page Accesses by Proxy Server Log”, ACM Journal of World Wide Web, Vol. 5, Issue 1, 2002

[47] Cheng-Zhong Xu; Tamer I. Ibrahim. "Towards Semantics-Based Prefetching to Reduce Web Access Latency", Proceedings IEEE Computer Society Symposium, SAINT’03, U.S.A.

[48] Cheng- Zhong Xu; Tamer I. Ibrahim. "Semantics-Based Personalized Prefetching to Improve Web performance", Proceedings IEEE Conference on Distributed Computing System, U.S.A.

[49] Holger Kirchner; RetoKrummenacher; David Edwards and Thomas Rissel. “A Location-aware Prefetching Mechanism”, Project work at Distributed Information Systems Laboratory LSIR, 2004

[50] V. Agababov, M. Buettner, V. Chudnovsky, M. Cogan, B. Greenstein, S. McDaniel, M. Piatek, C. Scott, M.Welsh, and B. Yin. Flywheel: Google’s Data Compression Proxy for the Mobile Web. In Proceedings of NSDI, 2015

[51] The Chromium Projects. SPDY. https://www.chromium.org/spdy, 2015.

[52] mod pagespeed. http://www.modpagespeed.com/.

[53] W. Zhou, Q. Li, M. Caesar, and B. Godfrey. ASAP: A LowLatency Transport Layer. In Proc. of the International Conference on emerging Networking EXperiments and Technologies (CoNEXT), 2011.

[54] TCP pre-connect. http://www.igvita.com/2012/06/04/chrome-networking-dns-prefetch-and-tcppreconnect/.

[55] S. Radhakrishnan, Y. Cheng, J. Chu, A. Jain, and B. Raghavan. TCP Fast Open. In Proc. of the International Conference on emerging Networking EXperiments and Technologies (CoNEXT), 2011

[56] S. Lohr. For Impatient Web Users, an Eye Blink Is Just Too Long to Wait.http://www.nytimes.com/2012/03/01/technology/ impatient-web-users-flee-slow-loading-sites.html, Mar. 2012

[57] S. Souders. Velocity and the bottom line.http://radar.oreilly.com/2009/07/velocity-making-your-site-fast.html, July 2009

[58] Netravali, R., Mickens, J., & Balakrishnan, H. (2016). Polaris: faster page loads using fine-grained dependency tracking. In 13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16).

[59] A. Datta, K. Dutta, D. Fishman, K. Ramamritham, H. Thomas and D. VanderMeer. A Comparative Study of Alternative Middle Tier Caching Solutions to Support Dynamic Web Content Acceleration. In Proceedings of 27th International Conference on Very Large Data Bases (VLDB), September 2001.

[60] A. Datta, K. Dutta, K. Ramamritham, H. M. Thomas and D. E. VanderMeer. Dynamic Content Acceleration: A Caching Solution to Enable Scalable Dynamic Web Page Generation. In Proceedings of ACM SIGMOD International Conference on Management of Data (SIGMOD), May 2001.

[61] A. Iyengar and J. Challenger. Improving Web Server Performance by Caching Dynamic Data. In Proceedings of Usenix Symposium on Internet Technologies and Systems, December 1997.

[62] I. Zukerman, D. Albercht and A. Nicholson. Predicting Users’ Requests on WWW.In Proceedings of 7th International Conference on User Modeling, June 1999.

[63] H. Zhu and T. Yang. Cachuma: Class-based Cache Management for Dynamic Web Content. Technical Report TRCS00-13, Dept. of Computer Science, The University of California at Santa Barbara, June 2000.

[64] Z. Su, Q. Yang, Y. Lu and H. Zhang. WhatNext: A Prediction System for Web Requests using N-gram Sequence Models. In Proceedings of 1st International Conference on Web Information System and Engineering, June 2000.

[65] Z. Wang and J. Crowcroft. Prefetching in World Wide Web. In Proceedings of IEEE Global Telecommunications Internet Mini-Conference, November 1996.

[66] Souders, S. (2007). High performance web sites: Essential knowledge for frontend engineers. Farnham: O'Reilly.

[67] Souders, S. (2009). Even faster web sites. Sebastopol: O'Reilly.

[68] Sundaresan, S., Magharei, N., Feamster, N., & Teixeira, R. (2013). Characterizing and Mitigating Web PerformanceBottlenecks in Broadband Access Networks.

[69] E. Cohen and H. Kaplan. Proactive caching of DNS records: Addressing a performance bottleneck. In Symposium on Applications and the Internet (SAINT), pages 85–94, 2001

[70] A. Feldmann, R. Caceres, F. Douglis, G. Glass, and M. Rabinovich. Performance of web proxy caching in heterogeneous bandwidth environments. In Proc. IEEE INFOCOM, New York, NY, Mar. 1999.

[71] Palmer, J. W. (2002). Web site usability, design, and performance metrics. Information systems research, 13(2), 151-167.

[72] Yagoub K, Florescu D, Issarny V, Valduriez P. Caching strategies for dataintensive Web sites. In: Proceedings of the 26th international conference on very large data bases, May 2000

[73] Podlipnig, S., &Böszörmenyi, L. (2003). A survey of web cache replacement strategies. ACM Computing Surveys (CSUR), 35(4), 374-398.

[74] Arun Iyengar and Daniela Rosu. 2002. Architecting Web sites for high performance. Sci. Program.10, 1 (January 2002), 75-89.

[75] Adali, S., Candan, K. S., Papakonstantinou, Y., &Subrahmanian, V. S. (1996, June). Query caching and optimization in distributed mediator systems. In ACM SIGMOD Record (Vol. 25, No. 2, pp. 137-146). ACM. Chicago

[76] J. Challenger, A. Iyengar, K. Witting, C. Ferstat and P. Reed, A Publishing System for Efficiently Creating Dynamic web Content Proceedings of IEEE INFOCOM, 2000.

[77] J. Challenger, A. Iyengar and P. Dantzig A Scalable System for Consistently Caching Dynamic Web Data, Proceedings of IEEE INFOCOM’99

[78] G. Apostolopoulos, V. Peris and D. Saha, Transport Layer Security: How much does it really cost? Proceedings of IEEE INFOCOM, 1999

[79] Vakali, A., &Pallis, G. (2003). Content delivery networks: Status and trends. Internet Computing, IEEE, 7(6), 68-74.

[80] Ravi, J., Yu, Z., & Shi, W. (2009). A survey on dynamic Web content generation and delivery techniques. Journal of Network and Computer Applications, 32(5), 943-960.

[81] Sivasubramanian S, Pierre G, vanSteen M, Alonso G. Analysis of caching and replication strategies for Web applications. IEEE Internet Comput 2007;11(1): 60–6.

[82] Yagoub K, Florescu D, Issarny V, Valduriez P. Caching strategies for dataintensive Web sites. In: Proceedings of the 26th international conference on very large data bases, May 2000

[83] Fortino G, Mastroianni C. Special section: enhancing content networks with P2P, grid and agent technologies. Future Gener. Comp. Syst. 2008;24(3): 177–9

[84] Candan, K. S., Li, W. S., Luo, Q., Hsiung, W. P., & Agrawal, D. (2001, May). Enabling dynamic content caching for database-driven web sites. In ACM SIGMOD Record (Vol.30, No. 2, pp. 532-543). ACM.

[85] Best Practices for Speeding Up Your Web Site: http://developer.yahoo.com/performance/rules.html

[86] Jayashree Ravi, Zhifeng Yu, and Weisong Shi. 2009. A survey on dynamic Web content generation and delivery techniques. J. Netw. Comput. Appl. 32, 5 (September 2009), 943-960

[87] Web performance optimization: http://en.wikipedia.org/wiki/Web_performance_optimization

[88] For Impatient Web Users, an Eye Blink Is Just Too Long to Wait: http://www.nytimes.com/2012/03/01/technology/impatient-web-users-flee-slow-loading-sites.html?_r=2

[89] Akamai Report: http://www.akamai.com/html/about/press/releases/2009/press_091409.html

[90] Rempel, G. (2015). Defining Standards for Web Page Performance in Business Applications. Proceedings of the 6th ACM/SPEC International Conference on Performance Engineering - ICPE '15.

[91] Galletta, D.F., Henry, R., McCoy, S. and Polak, P. 2004.Web Site Delays: How Tolerant are Users?, Journal of the Association for Information Systems: Vol. 5: Iss. 1, Article 1.

[92] Hoxmeier, J.A. and DiCesare, C. 2000. System response time and user satisfaction: an experimental study of browser based applications. Proceedings of the Americas Conference on Information Systems, (Long Beach, CA, USA, 2000), Association for Information Systems, 140-145.

[93] Google Page Speed Insights and Rules: https://developers.google.com/speed/docs/insights/rules

[94] Kambhampaty, S, Modali, V.S, Bertoli, M. Casale, G , Serazzi, G (2005). Performance Modeling for Web based J2EE and .NET Applications, In Proc. of world Academy of Science, Engineering and Technology, V 8, Oct.

[95] Barker, T. (2014). High performance responsive design: Building faster sites across devices (1st ed.). O'Reilly Media.