Large Scale Web Page Optimization of Virtual Labs

Jatin Agarwal*, Utkarsh Rastogi†, Prateek Pandey*, Nurendra Choudhary*, Venkatesh Choppella*, and Raghu Reddy*
*International Institute of Information and Technology, Hyderabad
Email: jatin.agarwal, nurendra.choudhary@research.iiit.ac.in
Email: venkatesh.choppella, raghu.reddy@iiit.ac.in
Email: prateek.pandey@students.iiit.ac.in
†National Institute of Technology, Durgapur
Email: utkarsh.rstg@gmail.com

***Abstract*—We propose set of guidelines for virtual labs to improve end user experience based on analysis of experimental results. Virtual labs were designed and developed by people with different technical backgrounds based on their familiarity with the technologies and the complexity of virtual lab. As a result each lab developer decided to use his or her own set of tools and technologies leading to performance issues during execution of the virtual labs. Performance of a web application depends upon the content of the web page. Therefore, we did analysis on web pages of virtual lab and listed guidelines to enhance user experience.**

I. Introduction

Virtual Labs is a project initiated by the Ministry of Human Resource Development, Government of India, under the National Mission on Education through Information and Communication Technology. The project aims to provide access to laboratories (remote as well as via simulation) in various disciplines of science and engineering for students at all levels from under-graduate to research. Virtual labs have been implemented as a collection of web applications where in the user may run the lab from a client machine (desktop or mobile device) invoking the various services offered on the server side. Hence the execution of the lab depends on the ability of the server or the client to process the requisite actions using the resources available on the server side or the client side. These resources are called *Critical Resources* [1]. *Critical rendering path* is the chain of necessary events that occur to render web page on browser. Front-end developers use too many critical resources like JavaScipt files, CSS files and images to make a good user interface which becomes an overhead. Therefore, while optimizing web page for speed we should focus on [2] minimizing the size critical resources, minimizing the number of critical resources and minimizing the critical rendering path

II. Background

Initially when world wide web started most of the web pages where plain hyper text documents. Slowly people started sharing media content like images and audio on web and in past decade web pages got transformed into web application which depend upon on critical resources like CSS, javascript and media content like images, audio, video etc. Nowadays rendering of web pages depends on the following three major criteria: (1) Order in which critical resources are fetched, (2) Number of critical sources, and (3) Size of critical resources. Web gaints like Yahoo and Google have listed best practices for web developers to improve performance of web pages [3]. Yahoo **Yslow** [4] and **Google Pagespeed** [5] are well known tools that are capable of evaluating web pages' performance and providing suggestions to optimize web pages based on the best practices listed above. Pagespeed is also capable of optimizing web pages without changing semantics of a web page.

III. Approach

Our work is broadly divided into four major phases namely Data Collection, Data Visualization, Analysis of Data and optimizing web pages based on analysis. During data collection phase we first collected all the urls of virtual labs hosted on www.deploy.virtual-labs.ac.in. Then we collected *Yslow* reports for each web page using an automated script and phantomJs. After collecting all the reports we extracted scores for each rule from reports and stored it in a csv file. During visualization phase all data is visualized using an automated script indicating performance for each rule and also overall performance of web pages. Later we carried out analysis to optimize the web pages of virtual labs.

IV. Experimentation

In order to achieve our goals we carried out experimentation on two different set of urls:

1) set of 5000 urls of *www.vlab.co.in* without *pagespeed*
2) set of 9000 urls of *www.deploy.virtual-labs.ac.in* with and without *pagespeed*

For each of the above set of urls we generate *yslow* reports and accumulate scores in three different CSV files. Later statistics are visualized in the form of bar graphs for each csv file. We analyze the performance of www.vlab.co.in using obtained bar graphs in next section. We also studied comparison between performance of 8786 web pages of www.deploy.virtual-labs.ac.in with and without *pagespeed*.

V. Guidelines for virtual lab to improve Perofrmance and user expereince

1) We found that virtual lab servers are not configured to use 'gzip' compression which adds to network latency. So it is recommended to configure 'gzip' compression on server side settings.

978-1-4799-6489-5/14 $31.00 © 2014 IEEE
DOI 10.1109/T4E.2014.51

IEEE computer society

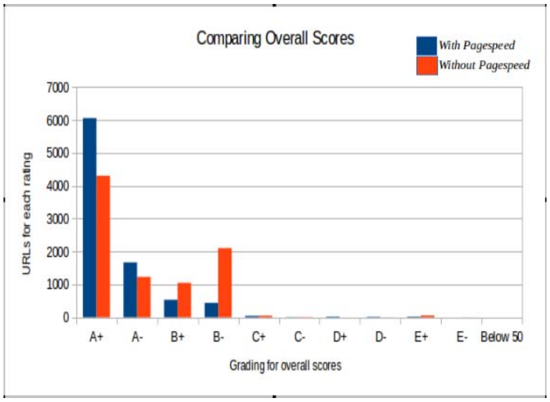

Fig. 1: Comparing Overall Scores

2) Most of the web pages of virtual lab makes lot of HTTP requests to the server which blocks the rendering. Therefore it is recommended to minimize HTTP request by using techniques like CSS spiriting, Image Maps, Combining scripts into one single file and inlining images.
3) We also recommend use of CDNs for huge web site like virtual labs with millions of users to enhance end user experience.
4) Most of the virtual labs web pages does not use an 'expire header'. Firstly we should follow 'Never expire' policy for static components by setting far future expire header. Secondly we should used cacheble control header for dynamic components. Both of the practices help us to efficiently browser cache and reduce number of HTTP requests in all subsequent visits to same web page.
5) Virtual lab pages does not use entity tags. Entity tags (ETags) are a mechanism that web servers and browsers use to determine whether the component in the browser's cache matches the one on the origin server. We recommend entity tags for the each component in virtual lab to enhance performance of a web page.

VI. Conclusion and Future Work

This paper is about Web Page Optimization of virtual labs in order to improve user experience. Our approach helped us in visualizing web performance and provide guidelines for virtual lab to improve performance. Our comparison study asserts that Google Pagespeed as strong utility for optimizing web applications. It should be used for huge websites like virtual labs, where various domains and backgrounds are involved.

This framework can be modified to give the list of urls of web pages that perform badly. Generating report takes at least 24 hrs to process 5000 urls but by making phantomjs clusters on framework like Hadoop we can parallelize the whole report generation. Also, the functionality of pagespeed can be improved by developing more filters. For example, there is no filter to give default favicon for a web page.

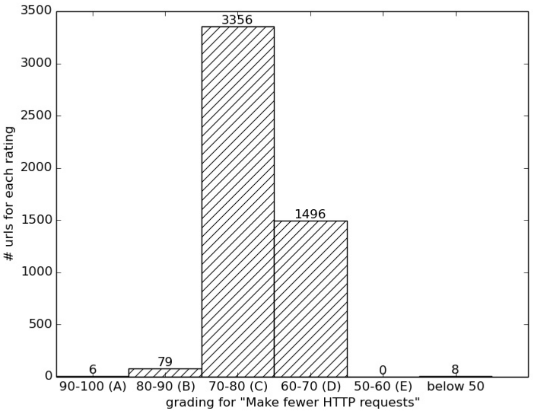

Fig. 2: Comparing Overall Scores

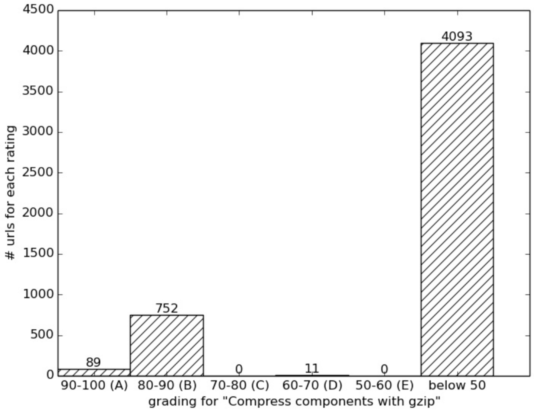

Fig. 3: Comparing Overall Scores

References

[1] Google Developers. Analyzing The Critical Rendering Path https://developers.google.com/web/fundamentals/performance/critical-rendering-path/analyzing-crp https://developers.google.com/web/fundamentals/perfor rendering-path/analyzing-crp

[2] Google Developers. Optimizing The Critical Rendering Path: https://developers.google.com/web/fundamentals/performance/critical-rendering-path/optimizing-critical-rendering-path https://developers.google.com/web/fundamentals/performance/critical-rendering-path/optimizing-critical-rendering-path

[3] Yahoo Developer Network–Best Practices for Speeding Up Your Website: https://developer.yahoo.com/performance/rules.html https://developer.yahoo.com/performance/rules.html

[4] Yslow Official Website.Available: http://yslow.org/phantomjs/http://yslow.org/phantomjs/

[5] Google Pagespeed Tools.Official Documentation: https://developers.google.com/speed/pagespeed/ https://developers.google.com/speed/pagespeed/