

# Monthly Report for August,2014

Jatin Agarwal

August 21, 2014

## Abstract

Last month we(Narendra and me) developed a framework to get *yslow* reports, hosted at <http://54.201.77.123/framework>. But *yslow* report's only list down issues related to client-side performance. But performance depends upon web page content (client side), networking and server-side parameters. So we explored *webpagetest* tool which take end into accounts almost all standard metrics and parameters while providing performance. *webpagetest* is an open source tool sponsored by Google for analyzing and finding existing trends in world wide web. So we were exploring *webpagetest* from past one week. We did install it on our local machine to start with it and later on the container. We used *npm* package to install it on our local machine. We read and understood it from following website <https://www.npmjs.org/package/webpagetest>. Later we explored multiple options provided to use it.

## 1 Introduction

WebPagetest is an open source project that is primarily being developed and supported by Google as part of our efforts to make the web faster. WebPagetest is a tool that was originally developed by AOL for use internally and was open-sourced in 2008 under a BSD license. The platform is under active development on GitHub and is also packaged up periodically and available for download if you would like to run your own instance. The online version at [www.webpagetest.org](http://www.webpagetest.org) is run by the WPO Foundation for the benefit of the performance community with several companies and individuals providing the testing infrastructure around the globe. We can run a free website speed test from multiple locations around the globe using real browsers (IE and Chrome) and at real consumer connection speeds. You can run simple tests or perform advanced testing including multi-step transactions, video capture, content blocking and much more. Your results will provide rich diagnostic information including resource loading waterfall charts, Page Speed optimization checks and suggestions for improvements.

The HTTP Archive crawls the world's top 300K URLs twice each month and records detailed information like the number of HTTP requests, the most popular image formats, and the use of gzip compression. We also crawl the top 5K URLs on real iPhones as part of the HTTP Archive Mobile. In addition to aggregate stats, the HTTP Archive has the same set of data for individual websites plus images and video of the site loading. Project was started in 2010 and got merged it into the Internet Archive in 2011. The data is collected

using WebPagetest. The code and data are open source. The hardware, mobile devices, storage, and bandwidth are funded by our generous sponsors: Google, Mozilla, New Relic, O'Reilly Media, Etsy, Radware, dynaTrace Software, Torbit, Instart Logic, and Catchpoint Systems.

## 2 Exploration

We explored most of the options provided by *webpagetest* and found them to be very useful for understanding whole performance issues with respect to a given url. It provides tones of options that suffices all our performance requirements for virtual labs. Webpagetest can be used in following tow ways. One option get API key from the admin of *webpagetest.org*. The API key allows around 200 requests per day on infrastructure provided by *webpagetest.org* and we have used this option for time being. The major advantage of using infrastructure provided by *webpagetest* is that we can do performance analysis from multiple locations and on all available browsers including internet explorer. Other we way is to host private instance of *webpagetest* on our infrastructure which will be expensive and complex affair. We propose this framework to be built using API's of *webpagetest* for measuring performance of virtual labs. We are developing set of shell scripts for building such an automated framework. Before we proceed we want to give a presentation on performance and *webpagetest* to get a roadmap for performance. So based on suggestion and improvements provided to us we would like to proceed further in developing an automated framework for measuring performance of virtual labs. Major advantage of using *webpagetest* is it defines metric for real time user performance measurement.

Last week I wrote an script that can be manipulated to send multiple requests to different locations and different browsers. Infrastructure provided by *Webpagetest* is sponsored by many different companies across the world. Setting such infrastructure require huge resources and cost. So before proceeding with our framework we want to know requirements of virtual labs. One solution is to capture web performance reports of all the existing web pages of virtual lab and store them in big database. So that lab developers can query this database to know performance of the web pages. Other advantage of this solution is that we can do large scale performance analysis of virtual labs. But in case of continuous development where web pages are modified frequently our model does not fit well. Main disadvantage of such a solution is that it is one time and requires lot of time for analyzing performance.

Another solution is to setup private instance of *webpagetest* triggered by continuous integration system like Jenkins and Travis. Therefore solution may require lot of resources in terms of time and effort needed to setup proper infrastructure. But once we setup such a system it would do continuous performance analysis on a periodic basis. Such setup would require dedicated infrastructure.

## 3 Proposal

We propose setting up our own instance of *webpagetest* on our own infrastructure. It would help developers in analyzing performance before proceeding to another experiment. It would also enforce best practices on lab developers and

can allow them to optimize web pages w.r.t performance. It will also act as a testing framework before allowing lab for deployment. This framework would be major step in software development life cycle because lab developers would receive continuous feedback and recommendations to improve labs w.r.t performance. We want to have detailed discussion on design of such a framework before proceed.

## 4 Conclusion

We propose that *webpagetest* should be used for analyzing performance based on requirements at virtual labs. Once we get performance results from *webpagetest* on existing status of all labs then we can start looking for solutions to improve performance. We can setup an continuous integration system like Travis and Jenkins on our infrastructure that does performance analysis of the labs periodically for lab developers.