Journal of Physics: Conference Series

**PAPER** • **OPEN ACCESS**

# Performance comparison of HTTP/2 for Common E-Commerce Web Frameworks with Traditional HTTP

To cite this article: Md Siduzzaman *et al* 2020 *J. Phys.: Conf. Ser.* **1529** 052023

View the article online for updates and enhancements.

# Performance comparison of HTTP/2 for Common E-Commerce Web Frameworks with Traditional HTTP

**Md Siduzzaman[1], Md Monir Hossan[1], Rakibul Alom[1], Talha Bin Sarwar[2] and Md Saef Ullah Miah[2]**

[1]Department of Computer Science, American International University - Bangladesh (AIUB), Kuratoli, Kuril, Dhaka, Bangladesh

[2]FSK1000 Fakulti Komputeran, University Malaysia Pahang, Gambang, Malaysia

talhasarwar40@gmail.com

**Abstract.** HTTP/2 is a cutting-edge Web convention predicated on Google's SPDY convention which tries to tackle the deficiencies and rigidity of HTTP/1. As e-commerce websites have become a significant medium for Online shopping, this paper demonstrates that whether HTTP/2 can authentically avail the performance of an e-commerce web browsing over HTTP or not. This paper states that we have studied about the HTTP/2 Implementation & performance analysis for prevalent web frameworks, where we have culled two different e-commerce web frameworks Laravel, WordPress (WooCommerce). At first, we have implemented two e-commerce sites in HTTP, then we have implemented those into HTTP/2. We additionally deployed them on the live server. By utilizing the Webserver Stress Tool & Selenium Web Driver, we have evaluated the performance under the sundry network environments. Selenium has given better results among all. But the webserver stress tool has exhibited some errors only for the http/2. This is our only constraint for this work. But still, we are endeavoring to resolve the problem. Simulation results have shown how HTTP/2 has influenced the page load time in our e-commerce websites. After analyzing all the simulation results, we have decided that Laravel is a more superior e-commerce web framework for both protocols, especially for HTTP/2.

## 1. Introduction

Hypertext Transport Protocol (HTTP) is a way of web browsing where web browser requests to a web server to ask for web pages and the other resources whatever they want. At first, HTTP/1 is developed & HTTP/2 is the updated & modified version of the HTTP protocol. HTTP/1allows only one request at a time, however, HTTP/1.1 adds request pipelining but this only partially addresses request concurrency and suffers a lot [3]. HTTP/2 is an accompanying advancement of HTTP that is being kept up by the IETF HTTP Working Meeting. Predicated on Google's SPDY tradition, HTTP/2 endeavors to act upon the insufficiencies of HTTP, focused on execution, like end-utilizer saw lethargy, framework, and server resources utilization [3][4]. With the work of more than two years' discoursed, 17 drafts and 30 completes, HTTP/2 and related HPACK subtleties were certified in February 2015 [4].

We, for the most part, plunge into the revealed inquiry in this paper, for example regardless of whether HTTP/2 helps web administrations execution for web-based business sites or not. Earlier work demonstrates that SPDY can either profit or now and again cost the PLT which is the page load time of Website pages when perusing credible webpages. We have found that HTTP/2 can enhance performance regarding page loading time [8]. Even HTTP/2 shows better performance than HTTP/1.1 regarding page downloading time (PDT) [6]. We are interested if HTTP/2 benefits to stacking online business sites with hindered abilities and more awful system. Regardless of the way that HTTP/2 could work except SSL, both Firefox and Chrome require the SSL [4]. On the other hand, most of the browsers support HTTP/2 well rather than HTTP [5]. In this way, we have thought about HTTP/2 with HTTPS instead of HTTP.

Both HTTP and HTTP/2 can be influenced by many elements outside to the conventions, including the system parameters (such as parcel misfortune, transfer speed), and highlights of Pages. Clients regularly look online business sites where heaps of items (information) subsists. We have separated our tests into two parts. Initially, we have looked at WordPress and Laravel web systems for an online business site that conveys loads of information. We have discovered that the Laravel system works so speedy and more secure structure as opposed to WordPress. At that point, we have analyzed HTTPS and HTTP as a transportation convention to exchange web articles for recognizing the components that may influence HTTP2 with transmission capacity, misfortune amount, number of elements on a page and articles' amounts. We have discovered HTTP costs high bundle misfortune, lower perusing speed than the HTTP/2. HTTP/2 learns progressively quick page loads, the life span of associations, more verified and increasingly economical Http ask for an internet business site. Additionally, this research attempts to answer the following two common questions:

- Q1: For what reason HTTP/2 helps or create problem under specific conditions?
- Q2: How does HTTP/2 perform after stacking extensive web-based business site?

The rest of the paper is sorted out as follows. The research background presents the foundation of HTTP/2. The methodology presents the estimation technique and how we have directed our investigations. Experimental setup and result analysis introduce the outcomes and examination of our analyses.

## 2. Research Background

In this segment, we exhibit the shortfalls as well as rigidities of HTTP and portray the corrections and highlights of HTTP/2.

### *2.1. Shortfalls and Firmness of HTTP*

HTTP has served the web for over a long time since the institutionalization. In any case, the web has transmuted a great deal to make it obsolete. As indicated by HTTP File [2], a site page is winding up increasingly unpredictable. It might take over 90 asks for more than 35 TCP associations with 16 distinct hosts to stack a web application, which possibly will exchange around 1.9MB of information all things considered. Nonetheless, HTTP essentially permits just a single exceptional demand for every TCP association, and HTTP is inactivity touchy and has consumed plentiful of the head of the line blocking issue [1]. Even though web designers have thought of various great practices like area sharing, spiriting, and in covering, the link of assets, these methods have their inadequacies [12].

### *2.2. HTTP2*

HTTP/2 gets after SPDY and indicates insufficiencies of HTTP. HTTP/2 keeps up alone, multiplexed association, supplanting the diverse affiliations per zone that programs disclosed in HTTP. HTTP/2 packs header data and transfers it in a minimized parallel structure instead of the plain substance approach. HTTP/2 uses one TCP connection. So, when the congestion window size of the connection has decreased no connection can use the network bandwidth [9][10][11]. The establishment of manifold connections also can be made in by higher latency and higher packet loss ratio.

Edge is an element of correspondence in HTTP/2 comprising an edge header that in any event observes the stream to which the edge has a spot. All HTTP/2 correspondence is being performed inside an affiliation that can pass on any number of bidirectional streams, which is each of a bidirectional stream of bytes. Along these lines, each stream passes on in messages which fuse one or unmistakable edges, which may all be interleaved and after that reconvened through the inserted stream identifier in the header of every individual bundling. There are three key highlights of HTTP/2: [13]

a) One association for every inchoation fundamentally decreases the related overhead: fewer attachments to oversee along the association way, littler memory impression, and better throughput. It moreover may prompt different advantages, for example, better pressure through the usage of a solitary pressure setting, less time in moderate begin, progressively quick blockage and misfortune recuperation.

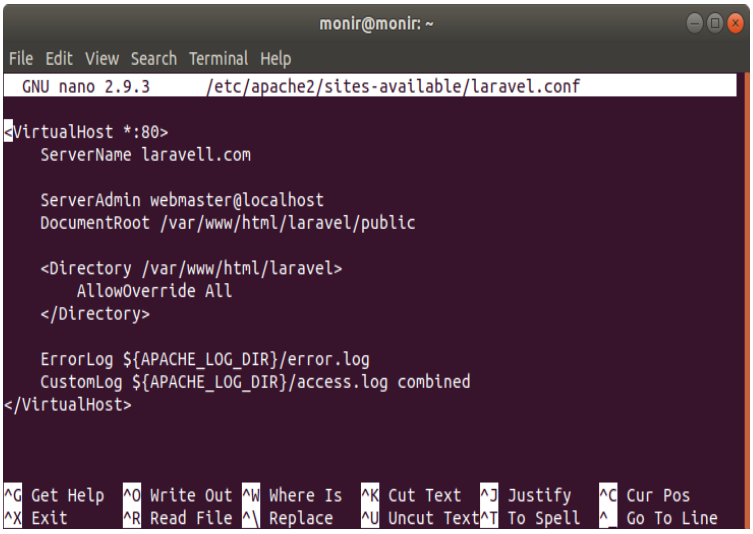
b) Demand prioritization. An HTTP message can be distributed to many separated frames. The precise request in which the edges are interleaved and conveyed can be streamlined for the further improvement of the execution. The browser can quickly post each demand whenever the asset is found, assign the need of each stream and let the server decide the ideal replication conveyance. This takes out unnecessary demand for lining inactivity and authorizations us to make the most proficient usage of every association.

c) Pushing the server. In extra to the reaction for the first demand, the server can push extra assets to the customer which can decrease the number of solicitations.

## 3. Experimental Procedure

To develop this work, we've picked two different web-based business structures named WordPress (WooCommerce) and Laravel.

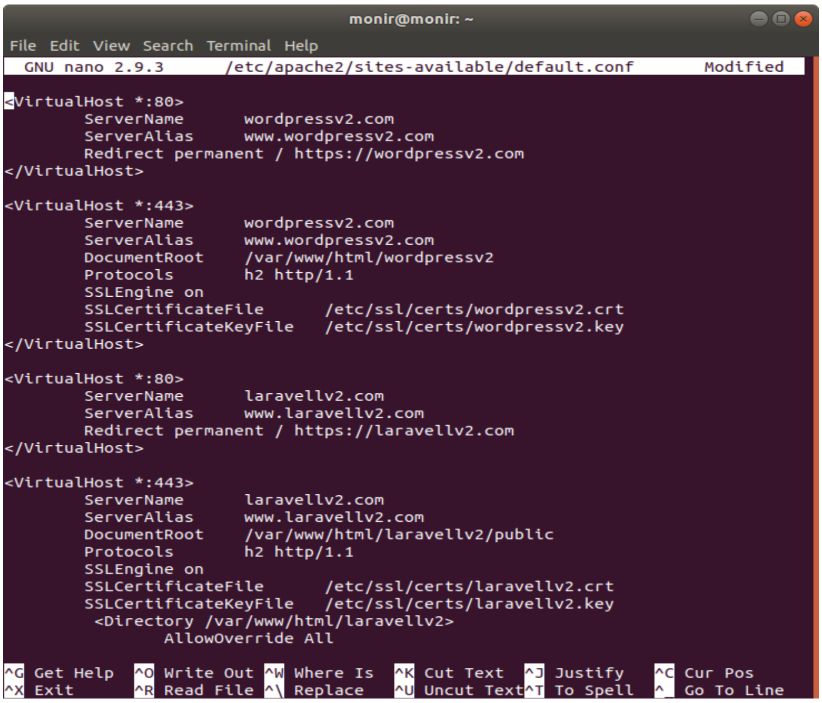
At first, we have developed two distinctive web-based business sites and hosted via HTTP protocol and then hosted them again over HTTP/2. For this, we have set up a virtual host in apache2 (LAMP Server). Figure 1 shows the setup of a virtual host in apache2 (LAMP Server) for HTTP.



```
  GNU nano 2.9.3       /etc/apache2/sites-available/laravel.conf

<VirtualHost *:80>
    ServerName laravell.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html/laravel/public

    <Directory /var/www/html/laravel>
        AllowOverride All
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>


^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Spell  ^_ Go To Line
```

**Figure 1.** Setting up a virtual host in apache2 for HTTP

Then, we have set up a virtual host in apache2 (LAMP Server) for HTTP/2. Figure 2 shows the setup of a virtual host in apache2 (LAMP Server) for HTTP/2.



```
  GNU nano 2.9.3      /etc/apache2/sites-available/default.conf       Modified

<VirtualHost *:80>
        ServerName      wordpressv2.com
        ServerAlias     www.wordpressv2.com
        Redirect permanent / https://wordpressv2.com
</VirtualHost>

<VirtualHost *:443>
        ServerName      wordpressv2.com
        ServerAlias     www.wordpressv2.com
        DocumentRoot    /var/www/html/wordpressv2
        Protocols       h2 http/1.1
        SSLEngine on
        SSLCertificateFile      /etc/ssl/certs/wordpressv2.crt
        SSLCertificateKeyFile   /etc/ssl/certs/wordpressv2.key
</VirtualHost>

<VirtualHost *:80>
        ServerName      laravellv2.com
        ServerAlias     www.laravellv2.com
        Redirect permanent / https://laravellv2.com
</VirtualHost>

<VirtualHost *:443>
        ServerName      laravellv2.com
        ServerAlias     www.laravellv2.com
        DocumentRoot    /var/www/html/laravellv2/public
        Protocols       h2 http/1.1
        SSLEngine on
        SSLCertificateFile      /etc/ssl/certs/laravellv2.crt
        SSLCertificateKeyFile   /etc/ssl/certs/laravellv2.key
         <Directory /var/www/html/laravellv2>
                AllowOverride All

^G Get Help   ^O Write Out ^W Where Is  ^K Cut Text ^J Justify   ^C Cur Pos
^X Exit       ^R Read File ^\ Replace   ^U Uncut Text^T To Spell  ^_ Go To Line
```

**Figure 2.** Setting up a virtual host in apache2 for HTTP/2

Then, we have set up the business sites which are two different copies of a same e-commerce site developed in Laravel and WordPress (WooCommerce) in HTTP/2 protocol. Figure 3 shows the website in HTTP/2 protocol.

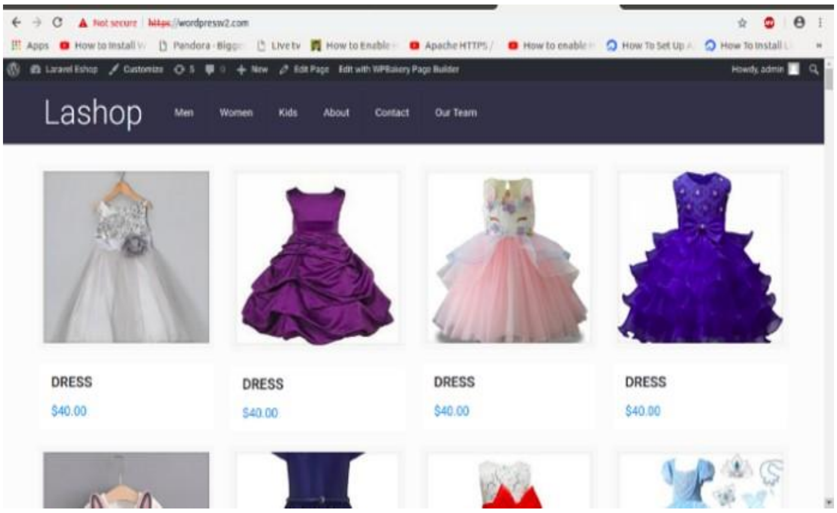
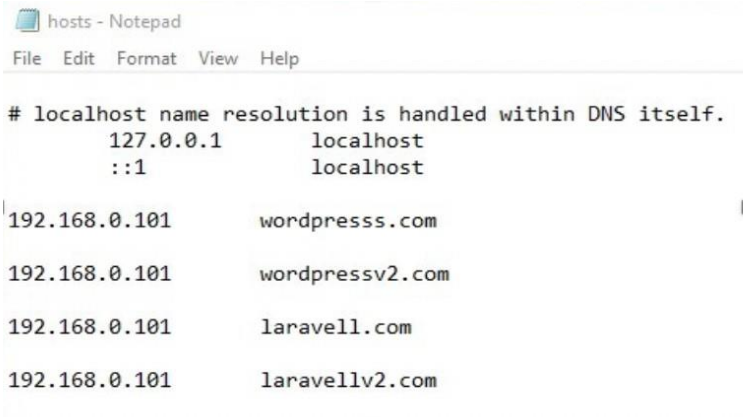


**Figure 3.** View of E-commerce site in HTTP/2 protocol.

Then, we have configured the Lamp server to access our virtual hosted website from the same network. Figure 4 shows the LAMP server configuration.



```
# localhost name resolution is handled within DNS itself.
	127.0.0.1       localhost
	::1             localhost

192.168.0.101       wordpresss.com

192.168.0.101       wordpressv2.com

192.168.0.101       laravell.com

192.168.0.101       laravellv2.com
```

**Figure 4.** Lamp Server configuration

And finally, we have installed Selenium [14] and Webserver Stress Tool for testing websites performance [7].

## 4. Experimental Setup and Result Analysis

For the outcome examination, we have utilized a Webserver Stress Tool. At first, we have arranged a host pc where our web ventures were set up. At that point, we have attempted to get to our web ventures from a differently configured client machine. Here, we performed a run time testing for 2 minutes. The number of clients was 300 and there were 5 seconds of deferral between snaps. The following tables, Table 1 and Table 2 show the host PC configuration and client PC configuration respectively.

**Table 1.** Host PC Configuration Processor

| Processor | Ram |
|---|---|
| Corei5 (Intel) | 4GB |

**Table 2.** Client PC Configuration

| Processor | Ram |
|---|---|
| Corei5 (Intel) | 4GB |

### *4.1. Range of Snap Times*

The following graph in Figure 5 demonstrates the dispersion of clients sit tight occasions for each keep running in the test. It represents the wait time per user to complete a click under the system load. Here x-axis represents user wait time, the y-axis represents the percentage of the user and the z-axis represents total time spent since the start of the test.

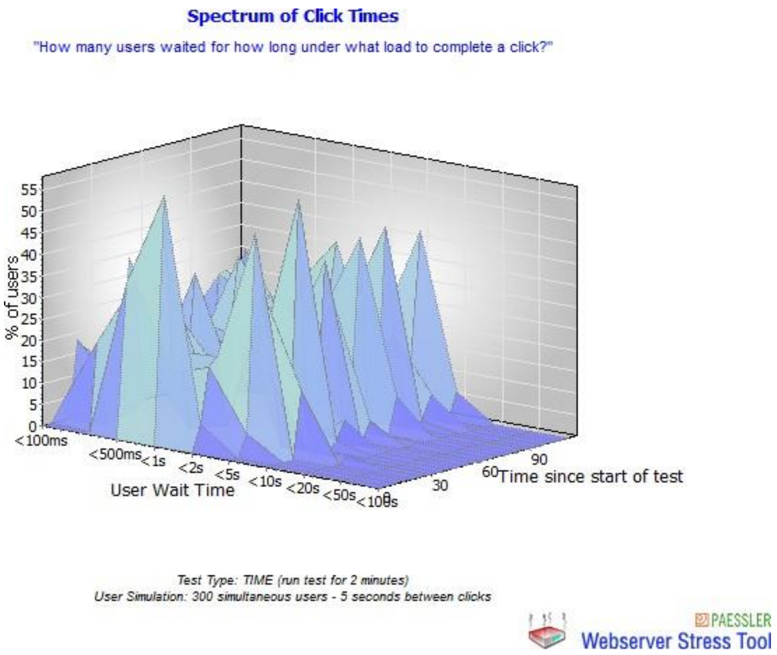


**Figure 5.** Spectrum of Snap Times

Figure also demonstrates that, toward the start of the test (first bars at the front of the diagram), most clients get to ask for times underneath 2 seconds, which means with the runtime of the test, the more users and load time increase, users face more wait time to get a job done. Here a click on any link and get the response from the click is mentioned as a job.

### *4.2. Server and Client Data Transfer Capacity*

The following graph in Figure 6 shows the data transfer capacity the server could convey (as an aggregate) just as the normal transmission capacity that was experienced by the mimicked clients.

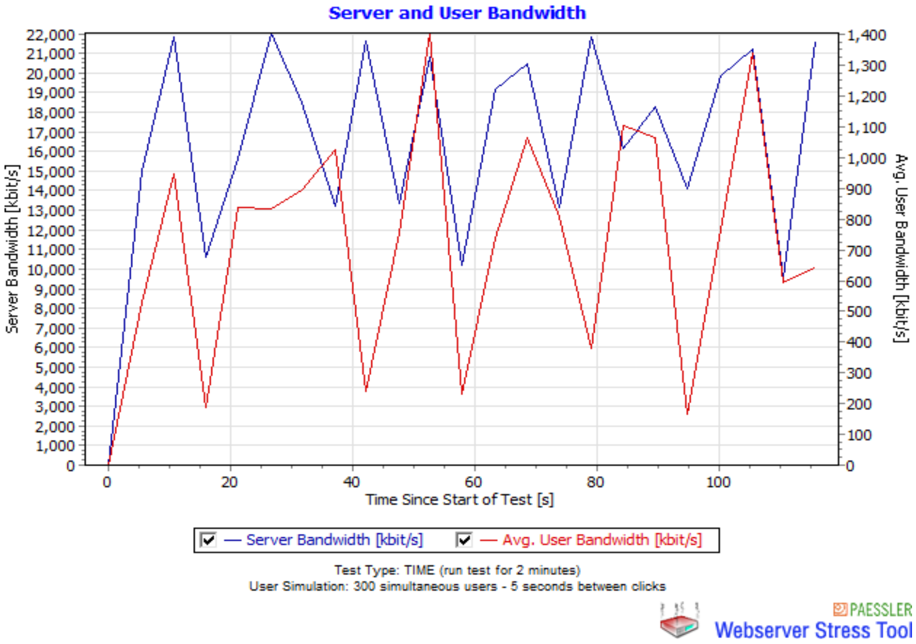


**Figure 6.** Server & User Bandwidth Time test

Figure 6 describes the server bandwidth utilization with time goes on while there is a continuous user request on the server. It demonstrates that the normal data transfer capacity accessible per client goes down from 15 Mbit to 3 Mbit when the quantity of clients moves from 1 to 20 seconds from the test start time. The average user bandwidth fluctuates with the number of users increment. For this test, the total number of simultaneous users was 300.

### *4.3. Snap Times and Blunders (per URL)*

The following graph in Figure 7 can be considered as the most critical outline since it shows the normal occasions and the rate of blunders that the reproduced clients have encountered while downloading pages amid the test.

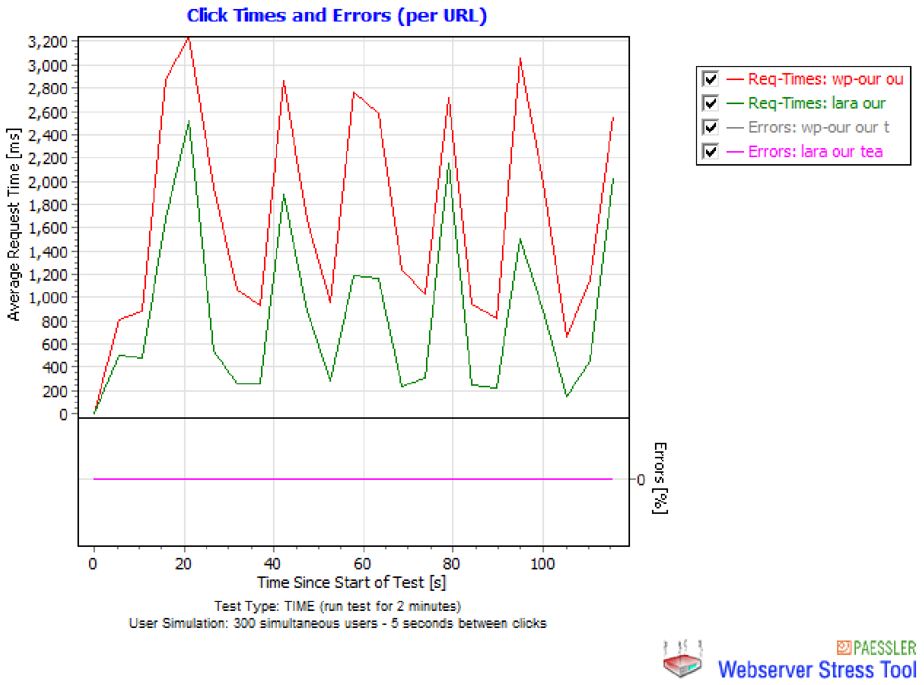


**Figure 7.** Snap Times & Errors (per URL) Time test

For every URL, the above graph demonstrates the demand times of snaps and the level of mistakes (in the lower some portion of the outline). Figure 7 demonstrates that the normal demand time for WordPress expanded up to 3200ms where 2400ms for Laravel. The most reduced normal demand time for Laravel was 180ms where WordPress utilized 700ms. Furthermore, the most considerable thing was that there were no mistakes for 300 synchronous clients as long as 2 minutes for time testing.

### *4.4. Snap Times, Hits/s and Snaps/s*

The following graph in Figure 8 shows the normal time for a client trusted that his demand will be prepared (counting diverts, images, frames, and so on whenever empowered), the hits every second and the clients per snap.

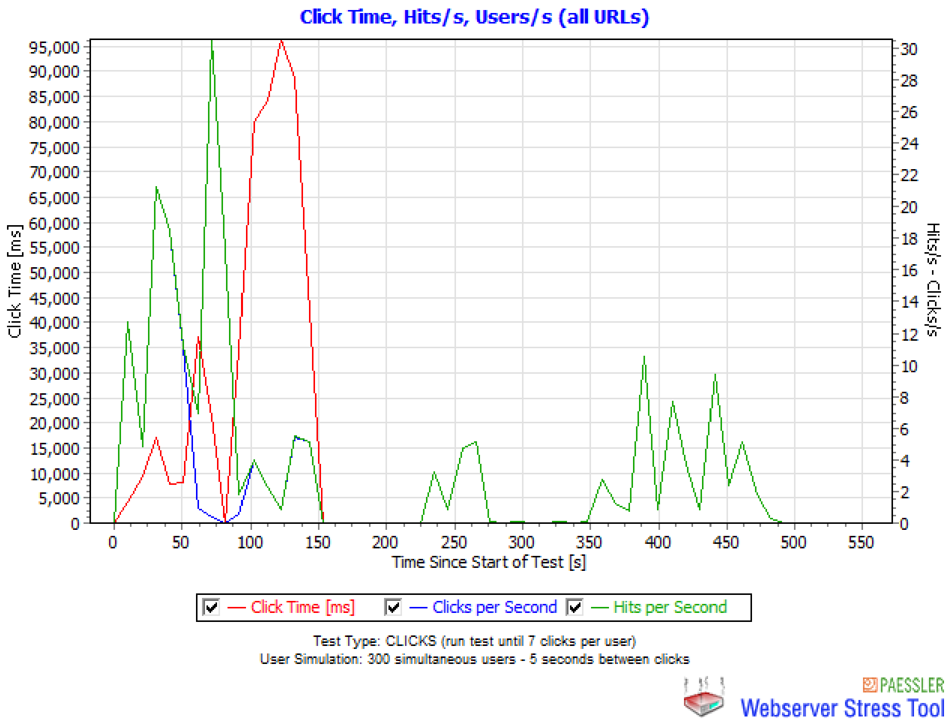


**Figure 8.** Snap Time, Hits/s, Users/s (all URLs) Time test

We can also see from the Figure 8 that for a certain time user can click or browse the site but after a certain time, 150s after the test started, the system stopped responding. Then again after some time, users request was there but users are unable to get any response. The maximum click per second the server can handle was 12 clicks and then the click time was 35000ms and hits per second was 18.

Our research work focuses to compare the performance of HTTP2 protocol for common e-commerce web frameworks like WordPress, WooCommerce, and Laravel web framework based e-commerce implementation. From our tests, we have found that in every case like user wait time per request while the users are increasing consistently. Laravel based implementation was working better than the WooCommerce one, even in the average bandwidth usage Laravel based implementation with HTTP2 was close to the server's bandwidth. From these tests, it is clear that Laravel's implementation with HTTP2 is better for basic e-commerce websites.

## 5. Conclusion

In this paper, we have contemplated the HTTP/2 execution and performance evaluation for the predominant web systems. From the test results, we have come to the conclusion that HTTP/2 gives better execution flexibility while using Laravel than E-commerce site developed using WordPress (WooCommerce). We have tested and evaluated the performance for different use cases. From there, we have found a better result for Laravel implementation with HTTP2.

## References

[1] Head of Line Blocking [Internet] [Cited April 2019]. Available from: https://en.wikipedia.org/wiki/Head-of-line_blocking

[2] Saverimoutou A, Mathieu B, Vaton S. Which secure transport protocol for a reliable HTTP/2-based web service: TLS or QUIC?. 2017 *IEEE Symp. on Computers and Communications (ISCC)* 2017 Jul 3 (pp. 879-884). IEEE.

[3] Hypertext Transfer Protocol Version 2 (HTTP/2). 2015 [Cited April 2019] Available from: https://http2.github.io/http2-spec/

[4] What is HTTP/2 – The Ultimate Guide. 2013 [Cited April 2019] Available from:

https://kinsta.com/learn/what-is-http2/

[5] The HTTP/2 Protocol: Its Pros & Cons and How to Start Using It. 2016 [Cited April 2019] Available from: https://www.upwork.com/hiring/development/the-http2-protocol-its-pros-cons-and-how-tostart-using-it/

[6] Corbel R, Stephan E, Omnes N. HTTP/1.1 pipelining vs HTTP2 in-the-clear: Performance comparison. 2016 *13th Int. Conf. on New Technologies for Distributed Systems (NOTERE)* 2016 Jul 18 (pp. 1-6). IEEE.

[7] Webserver Stress Tool: Free performance, load, and stress test for web servers. 2014 [Cited April 2019] Available from: https://www.paessler.com/tools/webstress

[8] Kim H, Lee J, Park I, Kim H, Yi DH, Hur T. The upcoming new standard HTTP/2 and its impact on multi-domain websites. 2015 *17th Asia-Pacific Network Operations and Management Symp. (APNOMS)* 2015 Aug 19 (pp. 530-533). IEEE.

[9] Yamaguchi S, Oguchi M, Kitsuregawa M. iSCSI analysis system and performance improvement of sequential access in a long-latency environment. *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*. 2006 Apr;**89(4)**:55-69.

[10] Yamaguchi S, Oguchi M, Kitsuregawa M. Trace system of iSCSI storage access. 2005 *Symp. on Applications and the Internet* 2005 Feb 4 (pp. 392-398). IEEE.

[11] Hayakawa A, Oguchi M, Yamaguchi S. Controlling middleware for reducing the TCP ACK packet backlog at the WLAN access point. *Management Studies*. 2017 May;**5(3)**:219-33.

[12] HTTP – Hyper Text Transfer Protocol. 2014 [Cited April 2019] Available from: https://www.w3.org/Protocols/

[13] Belshe M, Thomson M, Peon R. Hypertext transfer protocol version 2 (http/2). *Tech. rep., IETF*; 2015 May.

[14] Chandrasekara C, Herath P. Load and Performance Testing. *Hands-On Functional Test Automation* 2019 (pp. 219-248). Apress, Berkeley, CA.