



## Stress Testing for J-query Based M-Commerce Mobile Web Applications Using Webserver Stress Tool

**M. Asokan\***

Research Scholar & Associate Professor,  
PG and Research department of Computer Science,  
Nehru Memorial College, Puthanampatti, Tamilnadu, India

**Dr. P. Arul**

Assistant Professor, Department of Computer Science,  
Government Arts College,  
Tiruchirappalli, Tamilnadu, India

**Abstract**— *The growing embracement of smartphones, tablets and other mobile devices has fuelled the escalation of mobile websites in recent years. Mobile applications and mobile websites have become a major channel for conducting business, communicating, and reaching consumers. It is now mandatory for businesses to ensure the performance of applications in their mobile environments. The number of mobile devices has now surpassed the global human population. jQuery is an open source JavaScript library that simplifies the interactions between an HTML document, or more precisely the Document Object Model and JavaScript. jQuery is used to construct faster loading e-commerce /m-commerce websites. Webserver Stress Tool is a powerful HTTP-client/server test application designed to pinpoint critical performance issues in our web site or web server that may prevent optimal experience for our site's visitors. This paper discuss about the importance of jQuery based mobile websites and the need of stress testing for mobile websites.*

**Key words:** jQuery, M-Commerce, Response Time, Stress Testing, Virtual Users.

### I. INTRODUCTION

The m-commerce websites are the future of the world, growing at the faster rate. It requires a lot of investment, and it requires an extensive testing. Mobile commerce refers to the sale of goods through mobile devices – such as mobile phones or tablet computers – over a wireless internet connection. The m-commerce application should load fast so that the users get the best experience. Stress testing is a type of performance testing focused on determining an application's robustness, availability, and reliability under extreme conditions. The goal of stress testing is to identify application issues that arise or become apparent only under extreme conditions. These conditions can include heavy loads, high concurrency, or limited computational resources. By simulating the HTTP requests generated by hundreds or even thousands of simultaneous users, we can test our web server performance under normal and excessive loads to ensure that critical information and services are available at speeds our end-users expect. Webserver Stress Tool allows us quickly ascertain and identify performance problems so that we can quickly correct them to prevent user dissatisfaction and potential loss of revenue. Through an intuitive interface, flexible testing parameters, and comprehensive reporting, Webserver Stress Tool provides us the tool to include performance testing as a regular part of website and web application maintenance and deployment. The purpose of a stress test is to estimate the maximum load that our webserver can support. Webserver Stress Tool can help us to learn the traffic thresholds of our webserver and how it will respond after exceeding its threshold. This tool has been tested with a jQuery based leading m-commerce mobile website (<https://m.ebay.in>) for the three main test types (i) Clicks (ii) Time and (iii) Ramp.

### II. MOBILE APPLICATIONS

Mobile Internet will drive the next wave of growth in penetration of internet. Around 9 billion apps will be downloaded in India in 2015, more than five times the number of apps downloaded in 2012 (1.56 billion). Source: Deloitte India's TMT Predictions 2015 report. India's m-Commerce market to reach USD 19 billion by 2019.

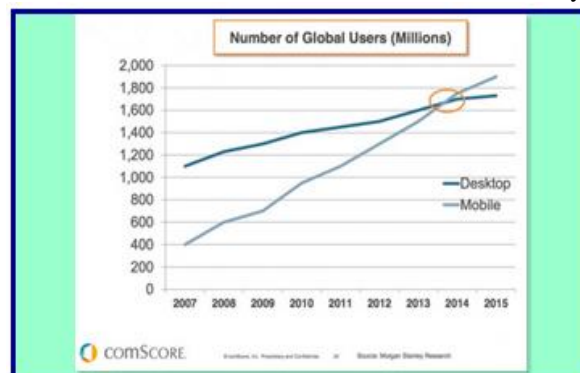


Fig-1

Mobile Apps Now account for more than Half (52%) of All Time Spent on Digital Media. Smartphone users spent 89% of their mobile media time using mobile apps ( <http://www.go-globe.com/blog/mobile-apps-usage/>). Mobile media time is now greater than desktop and this trend will continue. In 2008, marketers made a bold prediction: in 2014, mobile usage will overtake 'fixed' or desktop usage. Today, as evidenced by the chart shown in FIG-1 below, we see that this prediction has become a reality. According to a study by Google, 67% of people said they would be more likely to buy from a mobile-friendly site, and 61% said that a site that wasn't mobile-friendly would make them want to leave.

Mobile Applications are a rapidly developing segment of the global Mobile Market. A **mobile application or mobile app** is a piece of software that is designed for a specific mobile operating system and device type that can be downloaded by the user through a store like the Apple App Store or the Google Play store.

#### A) Types of Mobile Applications:

**Native Applications:** Native applications are developed for a specific platform and installed on the device.

**Hybrid Applications:** Hybrid application (hybrid app) is one that combines elements of both native and Web applications.

**Web Based Applications:** Web based applications are accessible through device browser or third party browsers installed on the device.

#### B) Benefits of a mobile site versus a mobile app:

1. Cost: A mobile site is typically less expensive to develop.
2. Time to Market: A mobile site will usually takes less time to develop.
3. Searchable: Like traditional websites they are searchable and therefore can be found by search engines and used with Search Engine Optimization tools.
4. Universal Availability: Once published the websites are immediately available on any mobile device, on any mobile web browser, unlike apps that must developed for each unique operating system and downloaded by the end user.
5. No Upgrades Required: When changes are made , they are published and are visible to all users and do not require a new download by each individual.
6. Shareable: Once a mobile site is found, it can be shared like any other link.

#### C) Mobile Application Testing:

Mobile application testing is a process by which application software developed for hand held mobile devices is tested for its functionality, usability and consistency.

### III. STRESS TESTING

Stress testing means simulating heavy load on the server to find the maximum number of users the server can handle. This number is also called a *crash point*. The crash point does not necessarily mean that server crashes or hangs. It can mean that errors start happening or that the server performance or response time fall below the level that your service-level agreement (SLA) defines. Proper stress testing is useful in finding synchronization and timing bugs, interlock problems, priority problems, and resource loss bugs. The idea is to stress a system to the breaking point in order to find bugs that will make that break potentially harmful. The system is not expected to process the overload without adequate resources, but to behave in an acceptable manner. Following are the types of stress testing: (i) Distributed Stress Testing (ii) Application Stress Testing (iii) Transactional Stress Testing (iv) Systemic Stress Testing (v) Exploratory Stress Testing.

#### A) Metrics for stress testing:

Metrics help in evaluating a System's performance and generally studied at the end of Stress Test. The following are the Commonly used metrics:

- ✚ Click : A simulated mouse click of a user sending a request (one of the URLs from the URL list) to the server and immediately requesting any necessary redirects, frames and images (if enabled).
- ✚ Request : An HTTP request sent to the server regardless of an answer.
- ✚ Hit : A completed HTTP request (i.e. sent to the server and answered completely). Hits can be the PAGE request of a "click" or its frames, images etc.
- ✚ Time for DNS : Time to resolve a URL's domain name using the client system's current DNS server.
- ✚ Time to connect : Time to set up a connection to the server.
- ✚ Time to first byte (TFB) : Time between initiating a request and receiving the first byte of data from the server.
- ✚ Click Time : The time a user had to wait until his "click" was finished (including redirections/frames/images etc.).
- ✚ Click Delay : The time a user needs to view the webpage he just downloaded until he initiates the next click
- ✚ User Bandwidth : The bandwidth a user was able to achieve.
- ✚ Sent Requests : Number of requests sent to the server during a period.
- ✚ Received Requests : Number of answers received from the server during a period.

#### IV. TOOL RESULTS AND DISCUSSION

Stress test should be run with maximum loads and even overload until the system crashes or its performance degrades, in order to measure bottlenecks. It is a good practice to start the test with good load for which application has been already tested. Then gradually increase the load to stress the system. The point at which the servers not responding to the requests have been considered as a break point/ bottleneck.

Webserver Stress Tool simulates anywhere from a few users to several hundred users accessing a website via HTTP/HTTPS at the same time. Based on a set of URLs or using a VBScript, the software simulates independent users requesting webpages from that URL including images, frames etc. Each user is simulated by a separate thread with its own session information. URLs can be parameterized for each user and the sequence of URLs can be varied. Webserver Stress Tool offers three main test types (i)CLICK (ii) TIME (iii) RAMP.

Table I Test types

CLICK	The test is finished when each user has initiated the given number of clicks. CLICK tests are the right choice to test specific URL sequences.
TIME	Tests that run for a specified number of minutes. A timed test is often used for “burn in tests”, e.g. to keep a server under full load for 10 hours.
RAMP	Ramp tests also run for a specified time, but with increasing load from 1 user to the specified number of users which is reached at 80% of test time. During the last 20% the full number of users is active. A Ramp Test is a great way to find out the limitations of your web server or web application.

We have tested this tool with the m-commerce and mobile web application <https://m.ebay.in/>. Each test type generates a detailed output about around 22 pages on the following topics.

##### A) Project and Scenario Comments:

Results per URL for complete test

URL#1 (): Average Click Time 32,390 ms, 30 Clicks, 2 Errors

URL#2 (): Average Click Time 12,140 ms, 76 Clicks, 1 Errors

URL#3 (): Average Click Time 8,779 ms, 91 Clicks, 0 Errors

URL#4 (): Average Click Time 8,506 ms, 43 Clicks, 2 Errors

Total Number of Clicks: 240 (5 Errors)

Average Click Time of all URLs: 12,354 ms

##### B) URLs to Test:

These are the list of URL's recorded during our test

##### C) Results per User

Table II Results per user

User No.	Clicks	Hits	Errors	Avg. Click Time [ms]	Bytes	kbit/s	Cookies
1	24	24	0	25,710	3,521,426	45.66	
2	44	44	0	12,615	8,643,082	124.58	
3	67	67	0	7,628	10,188,885	159.50	
4	40	40	0	11,986	1,316,934	21.97	
5	6	6	0	69,307	882,693	16.98	
6	24	24	0	15,083	4,808,549	106.27	
7	26	26	0	12,266	3,954,914	99.21	
8	5	5	1	52,894	152,202	4.60	
9	3	3	0	76,264	389,857	13.63	
10	11	11	0	15,244	2,201,850	105.05	

##### D) Results per URL:

Table III Results per URL

URL No.	Name	Clicks	Errors	Errors [%]	Time Spent [ms]	Avg. Click Time [ms]
1		30	2	6.67	906,912	32,390

2		76	1	1.32	910,490	12,140
3		91	0	0.00	798,865	8,779
4		43	2	4.65	348,747	8,506

The following are the *shortlisted graphs* obtained from our RAMP type test. These graphs are *dynamic*. We can enlarge a particular portion of any graph and this feature help us to identify the *errors and bottlenecks*. Check boxes allows the user to select/ deselect the particular option.

**E) Graph Click Times & Errors (per URL)**

This can be considered the most important chart because it shows the averagetimes and the rate of errors that the simulated users have experienced whendownloading pages during the test. For each URL, this graph shows the request times of clicks and the percentage of errors (in the lower part of the chart). If you enable “download images” there are a two more lines for each URL showing the average request times and errors for the images.

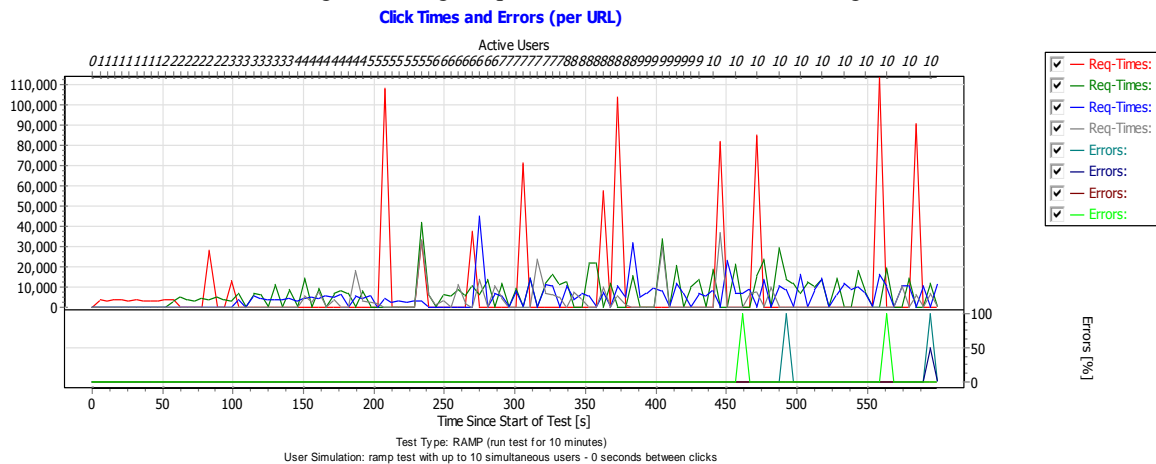


Fig .2 Click Times & Errors (per URL)

This sample graph shows the results of a 10 minute ramp test with up to 10users accessing four URLs of a webserver with 0 seconds between clicks.

**F) Graph Click Times, Hits/s and Clicks/s**

This graph shows the average time a user waited for his request to be processed (including redirects, images, frames etc., if enabled), the hits per second and the users per clicks. This time the values are calculated for all URLs together. The following graph shows the results of the same test as in the previous section:

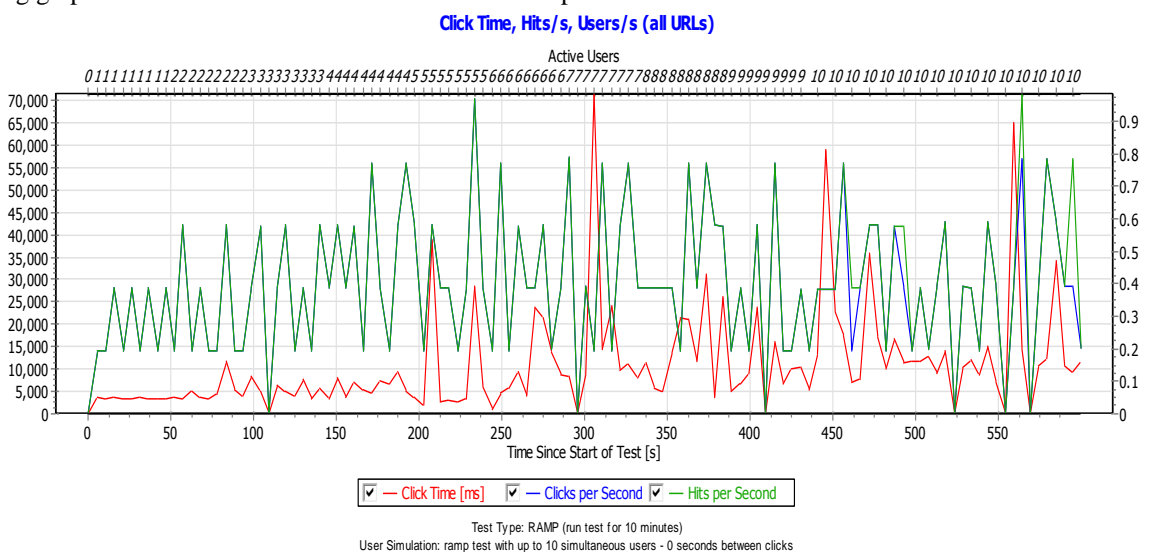


Fig. 3 Click Times, Hits/s and Clicks/s

**G) Graph Spectrum of Click Times**

This graph shows the distribution of user wait times for each run in the test.

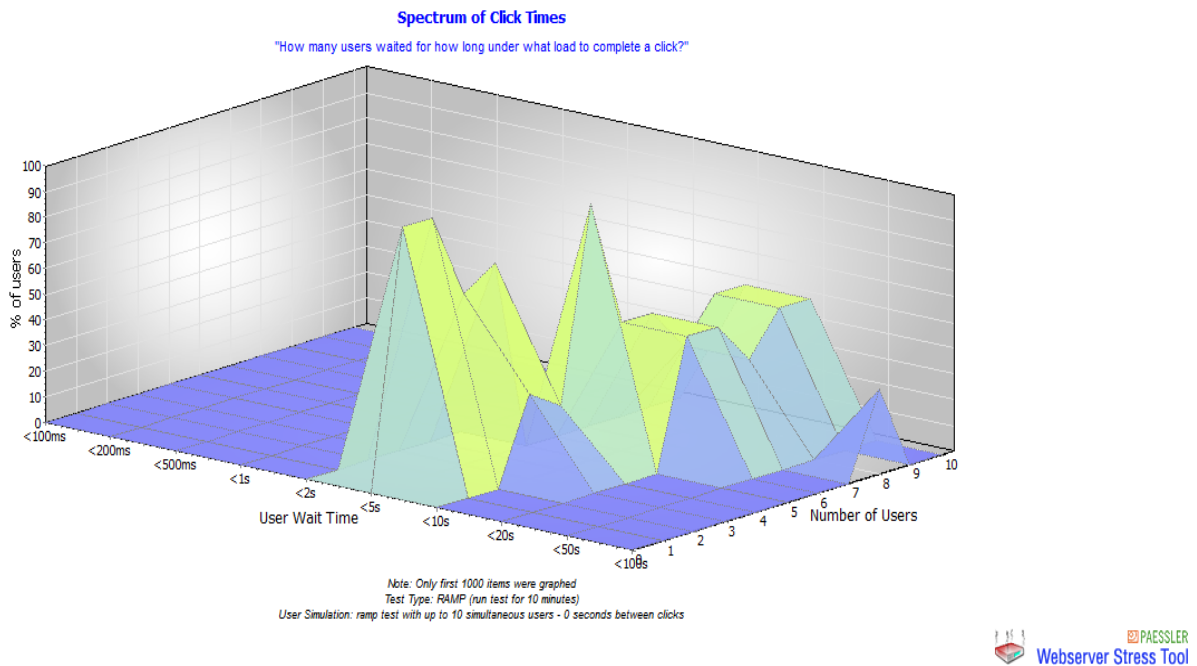


Fig. 4 Spectrum of Click Times

This sample graph shows the results of a Ramp Test. The three axes are:

- Vertical: percentage of users
- Horizontal: user wait time
- Depth: Number of users

At the beginning of the test (first bars at the front of the chart) most users get request times below 10 seconds. With more and more users accessing the server the request times deteriorate, the bar's maximum is moving from left to right with increasing depth.

**H) Graph Server and User Bandwidth**

This graph displays the bandwidth the server was able to deliver (as a total) as well as the average bandwidth that was experienced by the simulated users:

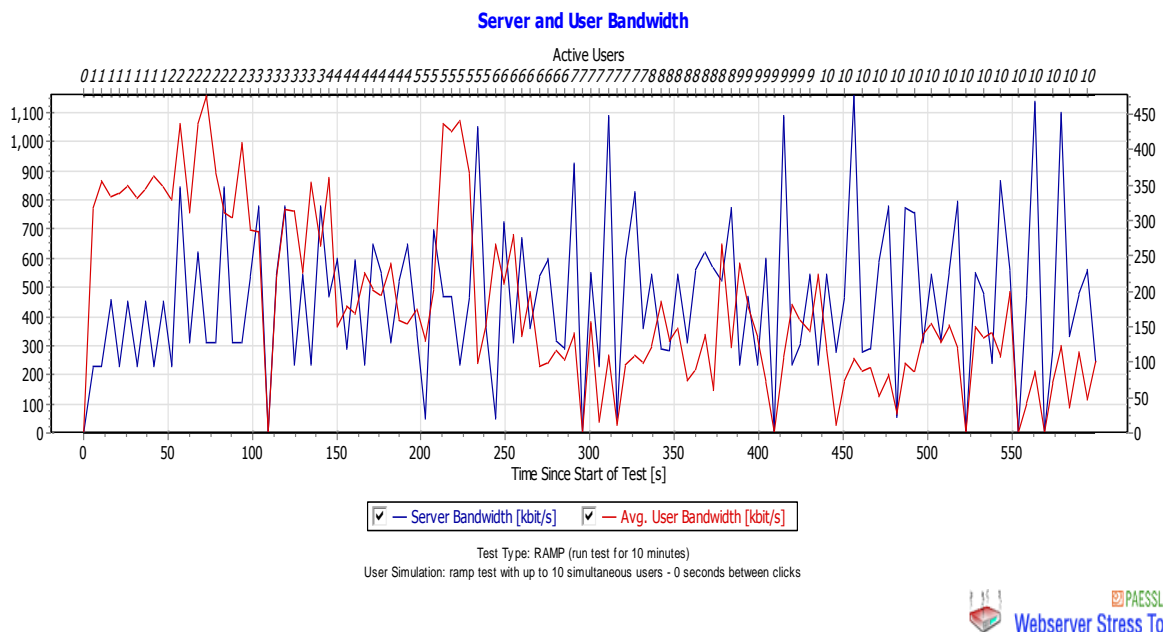


Fig. 5 Server and User Bandwidth

**I) Graph Open Requests and Traffic**

This graph shows the number of open requests as well as the number of sent and received requests in comparison with the network traffic:

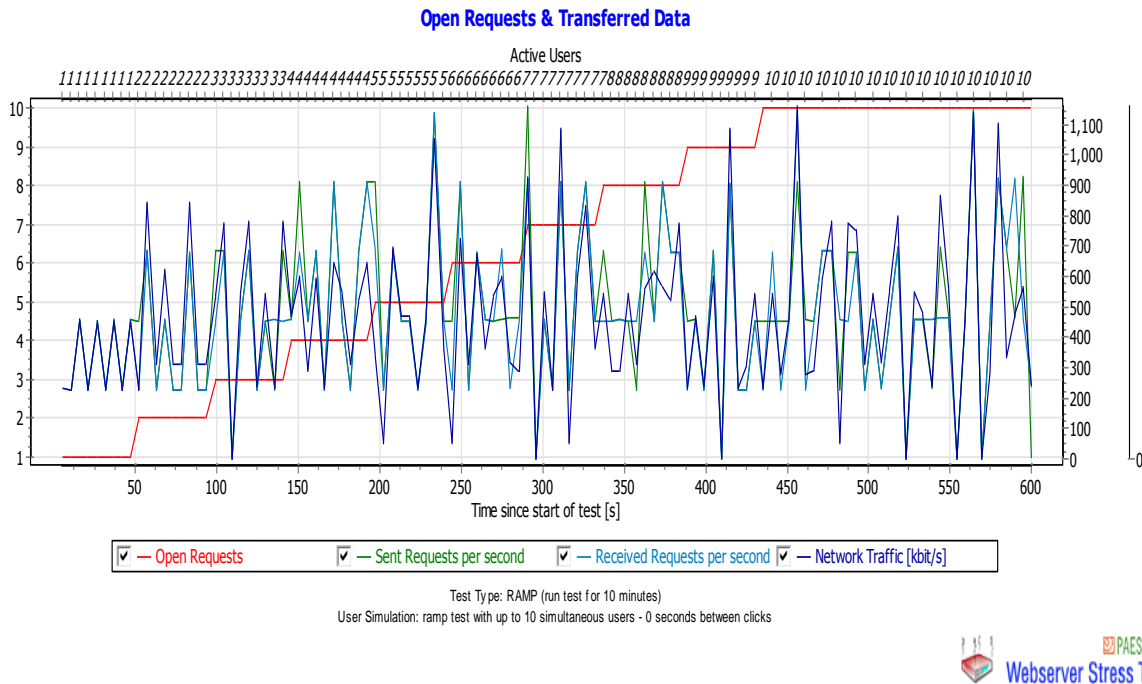


Fig. 6 Open Requests and Traffic

**J) Graph Protocol Times**

An HTTP request consists of several stages. First, the webserver name has to be resolved into an IP address using DNS (Time for DNS), then an IP port is opened on the server by the client to send the request header (Time to Connect). The server then answers the request (Time to First Byte) and sends all data. When all data is transferred, the request is finished (Click Time). Also in this graph a line is shown for the “time for local socket” which is the time that Webserver Stress Tool needed to acquire an open socket from the IP stack of the machine it runs on. For usual test this value should always be in the lower millisecond area (1-30 ms). For extreme traffic tests this value can rise above 50-100 ms which is a sign that the performance limits of the local machine have been reached. The average value of these five readings are displayed in this graph:

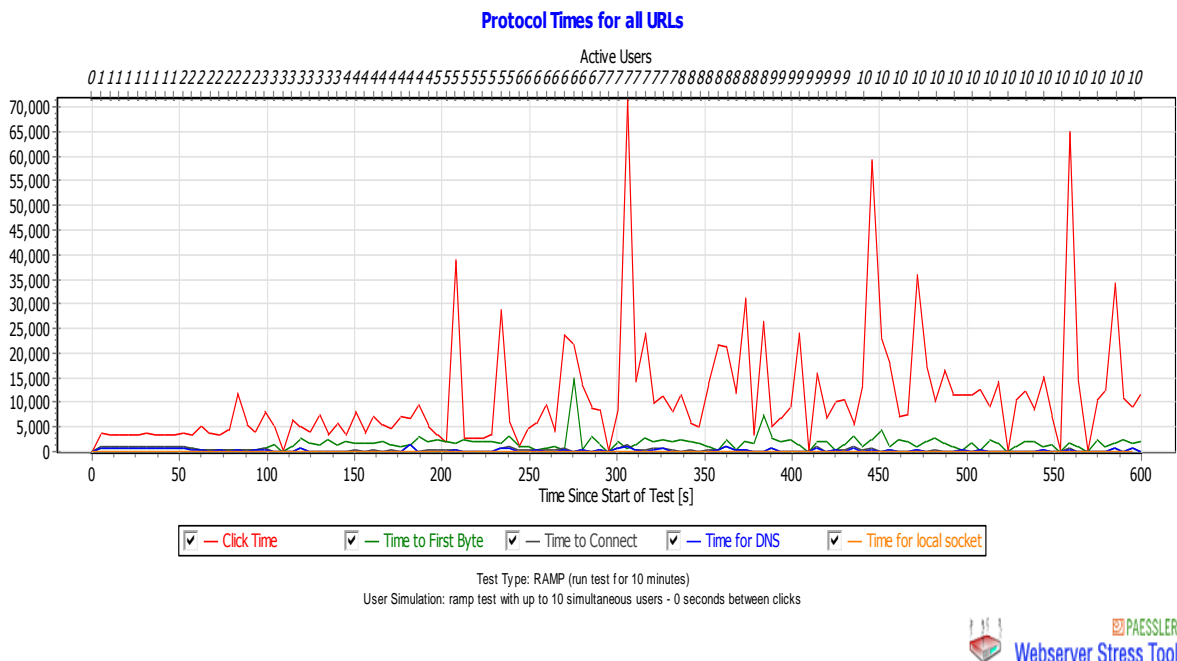


Fig. 7 Protocol Times

**K) Graph Test Client’s Health**

For this last graph Webserver Stress Tool constantly measures vital parameters of the machine it runs on. It can be helpful to find out if the limits of the test client have been reached. Especially the line for the CPU Load (pink) should be well below 100%. If we constantly hit values above 90% for the CPU load, the test results may be incorrect. Also the network traffic (blue line) should be below the physical limits of our connection to the server.

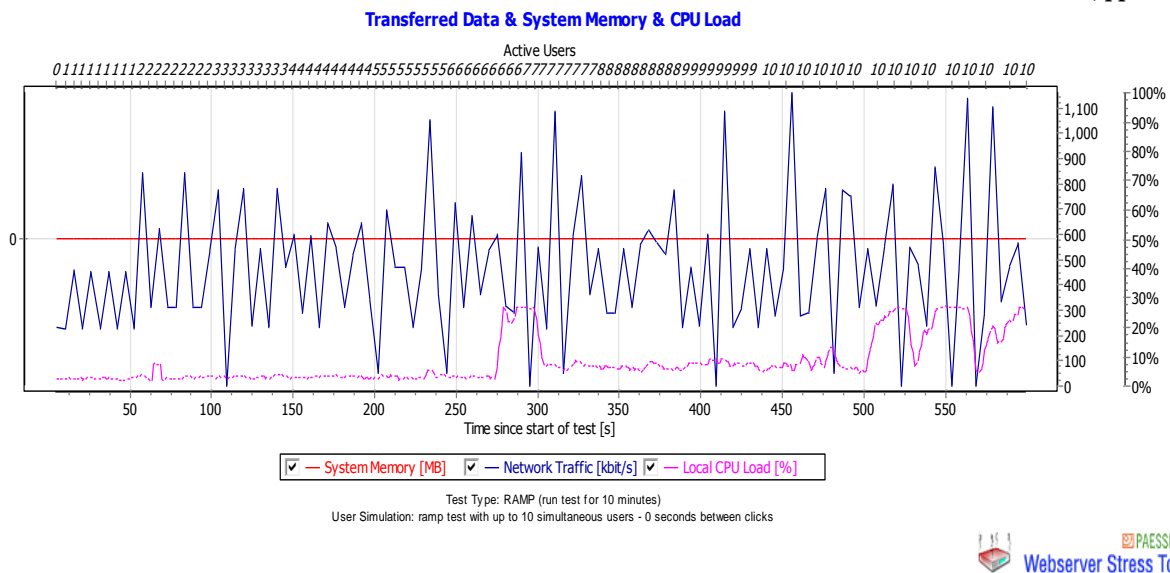


Fig. 8 Test Client's Health

## V. CONCLUSIONS

Through this paper STRESS TESTING FOR JQUERY BASED M-COMMERCE MOBILE WEB APPLICATIONS USING WEBSEVER STRESS TOOL has been presented. Necessary details about mobile web application usage trends , types of mobile web applications and a detailed description about mobile web application Stress Testing have been discussed. A popular jQuery based mobile website (<https://m.ebay.in/>) have been tested with WEBSEVER STRESS TOOL. The results have been presented with various graphs and analyzed.

## REFERENCES

- [1] Dr. P. Arul, M. Asokan, *Load testing for jquery based e-commerce web applications with cloud performance Testing Tools*, *International Journal of Computer Engineering & Technology (IJCET)*, 5(10) October (2014), pp. 01-10.
- [2] <http://docs.jquery.com>.
- [3] David Mercer (2012) *e-Commerce -How to plan, build, operate & market an E- Commerce store to make money online!*, Copyright ©David Mercer.
- [4] AsokanM ,*E-commerce websites with jQuery*, *International Journal on Computer Science and Engineering (IJCSE)*, Vol. 5 No. 03 Mar 2013
- [5] <http://www.indez.com/>
- [6] J.D. Meier, CarlosFarre, PrashantBansode, Scott Barber, Dennis Rea *Performance Testing Guidance for Web Applications*, © 2007 Microsoft Corporation.
- [7] <http://www.paessler.com/>