LOAD TESTING FOR JQUERY BASED E-COMMERCE
WEB APPLICATIONS WITH CLOUD PERFORMANCE TESTING TOOLS

Dr. P. Arul¹,   M. Asokan²

¹Assistant Professor, Department of Computer Science, Government Arts College, Tiruchirappalli-22, India
²Research Scholar & Associate Professor, PG and Research department of Computer Science, Nehru Memorial College, Puthanampatti, India

ABSTRACT

Online shopping is becoming increasingly popular because of speed and ease of use for customers. Online purchase intention rates have doubled since 2011 for 12 of 22 measured categories. Fractions of a second can make the difference between online success and failure. Faster websites and pages are proven to increase every business metric from revenue and cart size, to conversions and even search traffic. In today’s most competitive and rapidly changing web-based scenarios, it has become critical to the businesses and organizations to test their web-based applications thoroughly before they plan to launch their website. By performing rigorous testing of their web application, organizations can get an assurance that their applications will work smoothly and will be easily accepted by the end-users. This paper discuss about (i) E-Commerce website benefits(ii) categories of Performance/Load testing tools. A leading jQuery based E-commerce website (www.amazon.in) have been tested with cloud based performance testing tools Blazemeter and Load Impact and the results were analyzed.

Keywords: E-Commerce, Jquery, Load Testing, Response Time, Virtual Users.

I. INTRODUCTION

Load Testing is simply the process of subjecting a component or a whole system to a work level approaching its limits. Load testing is performed to determine a system’s behaviour under both normal and anticipated peak load conditions. The objective of Load testing is to determine (i) The speed with which a site under load will respond from the end-user’s perspective (ii) The maximum...
operating capacity of an application. (iii) The specific bottlenecks or infrastructure limitations that lead to poor website performance. Response time is a measure of how responsive an application or subsystem is to a client request. According to a survey of 160 organizations by Aberdeen Group, on an average a 1-second delay in page load time resulted in 7% loss in conversions. The loading time of a website is important, especially due to the following: (i) It’s one of the factors that search engines take into account in their algorithm results presentation. An excessive load time will affect the position. (ii) The speed of a website is one of the most important elements to consider usability. A website that loads slowly will cause the visitors to leave immediately. This will quickly result in fewer sales and conversions.

II. E-COMMERCE WEBSITES

An e-commerce website, is an online shop, a virtual space for online sale or where the company can expose its catalog of products and/or services, and conduct commercial transactions safely and without limits, with or without immediate payment, on this new channel of distribution and communication: the Internet! The most common E-commerce applications are Retail and Wholesale, Marketing, Finance, Manufacturing and Auctions. E-commerce website is very beneficial for the product sellers and service providers. Customers can directly buy the product from any website by online payment, credit card or debit card, the products will be dispatched to their given address. India’s E-Commerce market was about $2.5 billion in 2009, it went up to $6.3 billion in 2011 and to $16 billion in 2013 and is expected to grow huge $56 billion by 2023 that would be 6.5% of the total retail market. 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 websites. The following are some of the benefits of e-commerce on both a business and consumer’s perspective:

- Eliminate barriers of time and distance for a consumer to complete a transaction;
- Lower direct cost-of-sale over the internet in comparison to a conventional cost-of-sale with paper-based human interaction;
- Increased feasibility of selling niche products;
- Reduced cost of doing business;
- Decreased amount of time spent in resolving any discrepancies in orders or invoices;
- Reduced time and manpower needed to complete processes;
- Reduced pressure on certain resources.

III. LOAD TESTING TOOLS

WHY LOAD TEST?

A large spike in users can easily crash an unprepared website, causing lots of frustrated customers and huge losses in revenue. By load testing, web developers are able to see the problems ahead of time then properly prepare and adjust the site before it causes a problem for real customers. The load testing tools are classified into three distinct categories. (i) Freeware Load Test Tools (ii) Commercial Load Test Tools (iii) Cloud Based load Test Tools. Every toll has its own merits and demerits. Cloud Testing is a means of testing cloud-based applications that use resources (hardware, software and infrastructure) found in the cloud. Cloud testing provides an end-to-end solution that transforms the way testing is done and can help an organization boost its competitiveness by reducing the cost of testing without negatively impacting mission-critical production applications. This paper discuss about Cloud based Performance testing tools: Blaze Meter, and Load impact.
(I) BLAZEMETER

BlazeMeter can run multiple load tests that easily simulate load of up to 300,000 concurrent users from both the public cloud or inside the corporate firewall, enabling its customers to quickly locate and fix performance bottlenecks. We can create proprietary test scripts and load scenarios using a graphical web environment. BlazeMeter offers web-based test management, archiving, repository, cloud-based monitoring, rich scripting language, and supports HTTP/S, web-services, XML, TCP, SQL, Login (Flash, images, streaming) and more. BlazeMeter enables us to write load test-scripts using JMeter and user-experience test-scripts using Selenium. BlazeMeter will generate a load based on the JMeter script. The Selenium script is used during the load to automate the launch of real browsers to measure the real end-user experience. The load and monitoring is using a pre-configured distributed load testing environment. The environment is ready to use and available at all times. If we are not familiar with Selenium and do not wish to create a Selenium test-script, the system can generate one for our programmatically based on landing pages we provide. If we want the same for your JMeter script, BlazeMeter generates JMeter scripts automatically. With BlazeMeter, all we need is to write the test-scripts, choose the amount of load-engines and run the test. The system takes care of the everything else. Unlimited number of load-engines are pre-configured and available at our disposal. Detailed graphical reports are generated during the load.

(II) LOAD IMPACT

With Load Impact, we can quickly create a load test which simulates different browsers, network conditions and visitor behaviours – giving us the most realistic testing and the most reliable results. Record real user scenarios using their easy-to-use Chrome Extension or generate a load script automatically with just one click. For advanced testers, create custom scripts using our programmer-friendly IDE with code completion, JSON and XML parsing, data parameterization and more. Load impact helps the website owner/web master all of the following. (i) Ensure the response times experienced by your users are acceptable and within the boundaries of Service Level Agreements (SLAs) (ii) Validate server scalability and measure end user performance (iii) Establish a long-term baseline for your performance on which to validate the impact (good and bad) of infrastructure changes over time (iv) Determine the maximum capacity in order to understand which components fail first, in what order and what the nature of that failure is. (v) Locate and fix performance bottlenecks in the website’s backend by monitoring servers during testing.

IV. TOOL RESULTS AND DISCUSSION

A leading jQuery based E-commerce website (www.amazon.in) have been tested using the cloud based Performance/Load testing tools (i) Blazemeter and (ii) Load Impact. The results have been analyzed and also a detailed explanation for the various graphs have been presented.

(A) RESULT GRAPHS OF BLAZEMETER

Load Results: This report is a result of spawning numerous servers in the cloud, executing a JMeter test-script using distributed JMeter architecture. All graphs represent KPIs (Key Performance Indicators) measured during the load.
ACTIVE USERS shows how many virtual users are currently active. MAX USERS shows the maximum amount of active users. The difference between *active users* and *max users* is that the amount of active users will decrease when the test ends while the value of max users will remain the same. RESPONSE TIME is the amount of time the system takes to process a request after it has received one. This parameter does not include time that it needed to transmit HTTP/s requests over network to web-server under test. LATENCY is the delay involved for your request to reach the server. The *response time that is required to receive a response from the server is the sum of the response time + latency*. HITS/S is the quantity of HTTP/s requests per second that are generated by the test. KB/S is the average bandwidth consumption kilobytes per second that are generated by the test. 204 The server has fulfilled the request but does not need to return an entity-body, and might want to return updated meta information. The response MAY include new or updated meta information in the form of entity-headers, which if present SHOULD be associated with the requested variant.

ERRORS generated by the server during the test and errors due to connection timeouts, refusals or broken connections. 200 (Response code-ok) is the standard response for successful HTTP/s requests. Embedded Resources is the number of errors of embedded resources on the page. We can also use the slider at the bottom of the graph to change the scale of the graph.

The *aggregate report* creates a table row for each differently named request in our test. For each request, it totals the response information and provides request count, min, max, average, error rate, approximate throughput (request/second) and Kilobytes per second throughput. Once the test is done, the throughput is the actual throughput for the duration of the entire test. Samples are the number of samples with the same label.

Average is the average time of a set of results. Median is a number which divides the samples into two equal halves. Half of the samples are smaller than the median, and half are larger. [Some samples may equal the median]. This is a standard statistical measure. The Median is the same as the 50th Percentile. 90% Line (90th Percentile) meaning 90% of the samples took no more than this time. Median is the time in the middle of a set of results. 50% of the samples took no more than this time; the remainder took at least as long. Min is the shortest time for the samples with the same label. Max is the longest time for the samples with the same label. Error % is the percent of requests with errors. Throughput is measured in requests per second/minute/hour. The time unit is chosen so that
the displayed rate is at least 1.0. When the throughput is saved to a CSV file, it is expressed in requests/second, i.e. 30.0 requests/minute is saved as 0.5. Kb/sec - throughput measured in Kilobytes per second. Time is in milliseconds.

### FIG-2: AGGREGATE REPORT

**WATERFALL BREAKDOWN**

This report represents a waterfall breakdown of the traffic for a single user session (a single session = a single iteration), during the load period. Each bar represents the total time for a single user session - start-to-finish. This report is a result of automating Selenium. A real browser (Firefox) is launched numerous times during the load period, simulating the provided business process. Figures represent the page render times in a real browser.

### FIG-3: WATERFALL BREAKDOWN
ERROR REPORT
This report displays all errors received during the test run, categorized by labels (pages) and error types. The error generated during our test is shown in Fig- 4.

FIG-4: ERROR REPORT

MONITORING REPORT
Performance indicators received during the test while monitoring jmeter console and engines is shown in Fig-5.

FIG-5: MONITORING REPORT
(B) RESULT GRAPHS OF LOAD IMPACT

Chart: Chart plots the User load time and the number of Active clients in the same chart, creating a chart that gives us a rough idea of how fast the site is loading at various load levels. The user load time metric reports the amount of time a simulated client has spent loading resources from the tested server. If the load test is well prepared and executed, this metric will report the actual load time as experienced by a typical user on your site, at different load levels. Accumulated load time is the sum total of all individual load times, for all objects loaded by the client. It is a metric we can use to compare web server performance between load tests, as it gives a good picture of how fast things are loading in general, but it does not tell us “page load time” as experienced by a user.

FIG-6: CHARTS

The “Add graph” button allows us to add more metrics to the same chart. We also have a corresponding “Remove graph” button that removes individual graphs (metrics) from a chart. Clicking the “Switch view” button makes all charts half as wide so we can display two charts next to each other horizontally. The charting functionality is very powerful and allows us to combine different charts and graphs in almost any way we want. We can plot several different types of metrics in the same chart, for easy correlation between completely different metrics. There is also zoom functionality that lets you zoom in on different parts of the whole data set.

Below the charts we will find a list of pages found in our user scenario(s). A page can be seen as a bunch of URLs grouped together, forming what a visitor would experience as a page on the site.
When we want to know what individual URLs/resources the test has downloaded from our site, we can look at the URL list section. This, too, is collapsible and it shows every individual object/resource that was loaded from your server during the test. Like most data on the page, the URL list is updated dynamically throughout the course of the load test. Below is a screenshot showing only the first three URLs in an URL list section:

Along with each URL we get information about how many successful (where the server responded with a 200-response) and how many failed requests there were. Also, we get information about the lowest and highest load times seen for the particular URL, as well as the latest reported average load time. These statistics are reported per each URL, individual load zone, and individual user scenario, which means that we can e.g. check the load time for a certain URL and a certain load zone, or we can choose global statistics for a certain URL by selecting the results from “Aggregated, World”. If we have failed transactions for a URL, we can click the small (+) icon beside the URL to expand a more detailed statistics view for that particular URL. In that detailed view we also have a button that allows us to add a graph with results for the URL.

**Resource** type distribution is reported in a pie chart. The chart (Fig-9) shows how many of the resources loaded during the test falls into a certain category (such as “text/html” or “image/png”).
The final pie chart (Fig-10) shows the load time distribution for various resource types. That is how much of the total load time was a certain type of resource responsible for.

There are a lot of technologies behind a website to load. It is not necessary to use the same technology to all the pages of the website, mostly they are different. So the response time is also different. Google strongly suggests web-masters to get their web pages to load in under a second on mobile devices. This will ensure that people don’t abandon web pages due to inordinate delay.

Flipkart had deployed nearly 5000 servers and had prepared for 20 times the traffic growth. If the servers were different and are at different locations, the result metrics can’t be comparable. Both tools has its own way of presenting the results. Free Virtual user limits of Blazemeter and Load Impact are 50 and 250 respectively. The Chrome extension for Blazemeter and Load Impact simplifies the Load Testing process. Both the tools offers various pricing schemes for different number of Virtual Users. The website we have tested lies only within their respective free usage Virtual Users limits. Load testing has to be conducted for an expected number of users of the website under test. Each and every pages , various transactions and shopping carts has to be tested. Google Analytics may be used to find the expected number of users and their usage patterns. More over
since the users of the website spread all over the world it is necessary to run the test in servers at different geographic locations. Both tools provide options to run the same test at different locations. The goal of load testing any site is to increase the revenue. The budget and expected number of users are the only factors to find the appropriate load testing tool.

Sometimes users load the same site that they are load testing and experience a much faster load time than what is reported by the load testing tools. This could be due to a few possible reasons:

(i) User is on a network that is geographically closer to the server. If the website server is located far away from the load generators but close to the website user, traffic will have to take a longer route to reach the load generator. The extra network delay introduced, can affect test results greatly, depending on the composition of their site. A longer route also makes it more likely that there are bandwidth bottlenecks somewhere along the way. (ii) The web page can appear fully loaded graphically to the user but there might still be HTTP transactions executing in the background. The load generators report load times that take into account all HTTP transactions on the website. It is very important to maintain the response time within limits. If it exceeds the limits the website has to lose its customers and it results in loss in revenue.

We have listed some of the important points relevant to load testing:

(i) Load testing is a type of performance testing conducted to evaluate the behaviour of a component or system with increasing load, e.g. numbers of parallel users and/or numbers of transactions, to determine what load can be handled by the component or system. (ii) Every website has its own technology. (iii) Every Performance testing tool supports limited number of technologies. (iv) It is necessary to choose the best tool for the website being tested. (v) Obtaining permission from the website Administrator/Proprietor to run a Load test. (vi) It is mandatory to conduct the Load Tests periodically.

V. CONCLUSION

Through this paper Load testing with cloud based tools for jQuery based E-commerce web application has been presented. Some details about online sales statistics, advantages of e-commerce websites and a detailed description about Load Testing and importance of Load testing have been discussed. A famous leading jQuery based E-commerce website have been tested with Cloud based Performance/Load testing tools Blazemeter and Load Impact. The results for both the tools have been presented with various tables and graphs.

REFERENCES