



An Efficient Approach to Calculate Overall Response Time during Load Testing

Rahul Sharma

M.Tech Student, Department of CSE,
International Group of Institution, Sonipat
Haryana, India

Abstract—Most of the applications in the today's world are written as web-based application. These applications run on the browser like Internet Explorer, safari, Mozilla Firefox, etc. The load testing tools that are available in the market works only with the request and response i.e. they do not execute JavaScript at their end and hence does not measure the overall response experienced by the user. On the other hand, an Automation tool can measure the overall response time but we cannot generate load using an Automation tool. In this Paper, a new approach is used to measure the overall response time experienced by an end user. With this new approach, Overall response time will be successfully calculated during load test. The results show that new approach improves the software performance by providing us better results.

Keywords - Jmeter, selenium, web, Load testing using selenium, Performance Testing

I. INTRODUCTION

Load testing tools are used to test the application with multiple users. This is done by creating performance scripts for the different transactions (e.g. *Launch the application, Basic search, Advanced search, etc...*) which we want to test. The aim of the load test is to find the response time of the different transactions that will be experienced by the end user once the application will be live and to measure the server's efficiency and reliability. The load test tools works with the URL request only. It acts as a browser whose main task is to request a page from server by throwing URL to it. The server sends back the response to the tool. A tool can throw n number of requests for multiple users which is not possible manually for a single user to achieve. In this way the tool measures the response time taken by different transactions. But, are we missing something. Yes, because there may be delay that will be caused by the client side implementations (e.g. JavaScript). Although, that delay will be at the client end but it will be experienced by the end user. A load testing tool will not measure the time spent on client end processing by the browser as it does not launch browser while testing and hence does not measure the time taken at the client end.

On the other hand, an automation tool is used to test the functionality of the web application. Doing it manually, a human error can easily occur. [5] At the same, it wastes the time. Hence, there is the need for automation testing. The web based applications these days are having so much business logic that is not possible to catch all the issues manually. An automation testing tool is used to test the application by single user only. But, it is possible to use different browsers.

II. RELATED WORK

The idea of writing this research paper is to find an approach with which we can get the overall response time experienced by an end user during the load test. This paper provides an approach to measure response time by the integration of Load testing tool and Automation testing tool. [1]

The load testing tools do not execute JavaScript at their end and hence are unable to provide the overall response time experienced by the user. There are different tools available in the market that are used to test the client end processing but again these tools are highly paid and one needs a good skillset to use these tools.

III. ACTUAL APPROACH

The approach is to generate the load using a load testing tool and measure the response time using an Automation tool. The below points shows that how one can measure the overall response time taken using Automation testing tool which includes the Server processing time, Data transfer time and page rendering time[3]. Figure1 clearly explain the overall approach.

Overall Response time = Server processing time + Data transfer time + page rendering time

- Execute load test on the application using Load testing tool (e.g. Jmeter). This step is done to generate load on the application. But, we will not measure response time using this.

- Capture the response time of the different transactions using Automation testing tool (e.g. Selenium). The response time captured in this case will be the overall response time.

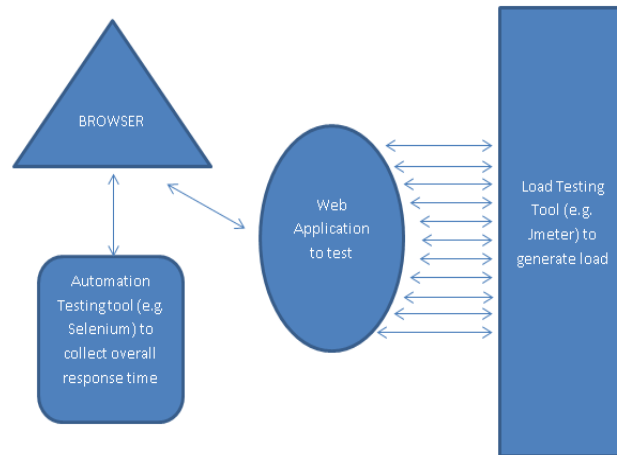


Figure 1. An approach to get Overall response time using automation tool during load test.

IV. IMPLEMENTATION

Here, we will implement the above approach using selenium (V 2.35.0) and Jmeter (V 2.10).

The code below[4] capture the current time before the transaction, then click on the search button (it may be any transaction) and then wait till it find the search results and then captures the current time again. Subtracting the finish time from the start time will provide us overall response time taken. Waiting for the search results is required as it will prove that the page is loaded completely.

```
long start = System.currentTimeMillis();
```

```
driver.findElement(By.id("Search")).click();
```

```
WebElement element = driver.findElement(By.id("ID of an element on the page which will load e.g. any search result"));
```

```
long finish = System.currentTimeMillis();
```

```
long OverallTime = finish - start;
```

```
System.out.println("Total Time for search - "+ OverallTime);
```

The Figure 2 shows the Jmeter tool [2] with scripts that are to be executed to generate load on the application. Execute the Jmeter scripts with the no of users for which application has to be tested and then run the selenium scripts to capture the overall response time. This response time will be the actual response time experienced by the end user.

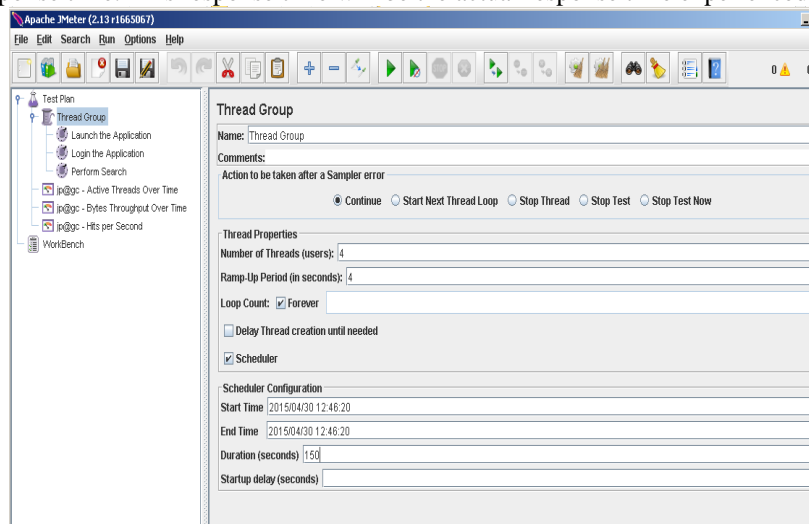


Figure 2: Jmeter tool

V. RESULTS

The Table 1 shows the actual results of the different transactions are captured by selenium during the load test.

Table1: Response Time of Transactions Captured by Selenium

S.No.	Transaction name	Time taken (in ms)
1	Launch the application	3256

2	Login the application	4698
3	Total Time for search	3564

Figure 3, 4 and 5 shows that graphs that were captured using Jmeter after the load test. These graphs are very much required for analyzing the load test results and to find the performance issues.

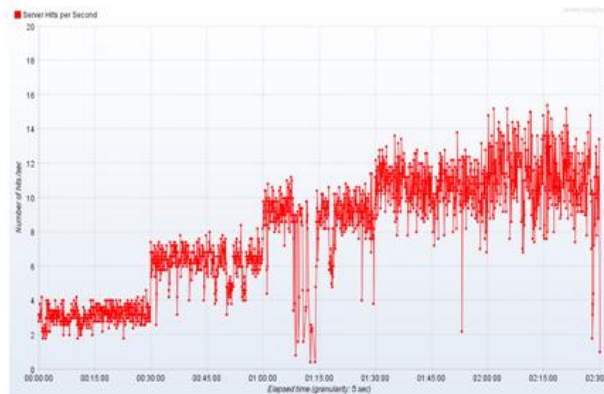


Figure 3: Server hits per second

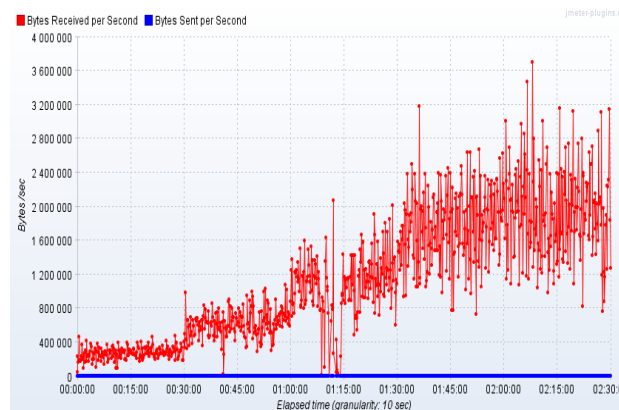


Figure 4: Bytes received per second

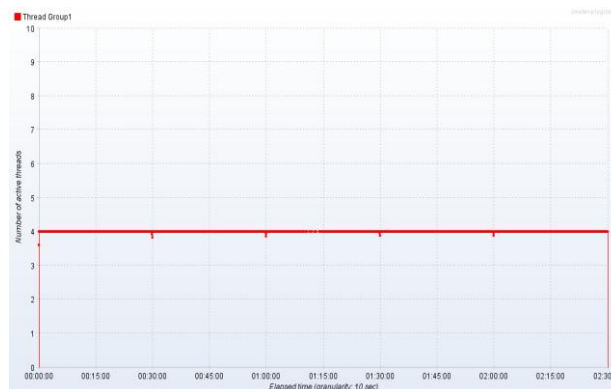


Figure 5: Active Threads over time

VI. CONCLUSION

The new Load testing approach integrated by Automation tool (e.g. Selenium) and Performance tool (e.g. Jmeter) can be easily used to capture the overall response time of the different transactions and to find any performance bottlenecks. The above approach is easy to use and provides us efficient results. Moreover, it possible to find the response time of different transactions on different browsers (IE, Mozilla Firefox, chrome, etc.) as we can execute the selenium scripts using different browsers. It can be widely used in web application testing.

REFERENCES

- [1] Fei Wang, Wencai Du: *A Test Automation Framework Based on WEB*, 2012 IEEE/ACIS 11th International Conference on Computer and Information Science
- [2] Jmeter website, <http://jakarta.apache.org/jmeter/>
- [3] <http://sqa.stackexchange.com/questions/6196/tools-to-measuring-page-rendering-time>

- [4] <http://stackoverflow.com/questions/11245062/how-can-we-get-exact-time-to-load-a-page-using-selenium-webdriver>
- [5] Ying Wu, *Automation Testing Framework for web base on selenium. Infomation Techonology[J]*, 2011,9:187-188