

A MODEL FOR PERFORMANCE TESTING OF AJAX BASED WEB APPLICATIONS

Bhupendra Singh¹, Shashank Sahu²

¹M.Tech Scholar (CSE), Ajay Kumar Garg Engineering College, Ghaziabad – UP- India

²Associate Prof. CSE Department, Ajay Kumar Garg Engineering College, Ghaziabad – UP- India

Abstract

The term “Web Application” (also known as Web App) is used to refer a program or application that is accessed through a network. There has been a shift in the direction of web development. A new breed of web application, known as AJAX is emerging in the field of web development. The aim is to make web page feel more responsive by exchanging small amount of data. In contrast with traditional web application, there is no need to reload web page entirely at each time when user make change. Ajax (Asynchronous Java script And XML) is a bundle of technologies used to simplify the implementation of rich and dynamic web application. AJAX web application consist a single page whose elements are updates dynamically and user can work asynchronously with such application.

To provide guidance to tester, for applying Ajax on traditional web application, we propose a model. Because most of the time tester think that performance will increase due to number of advantage of Ajax but it is not necessary that all the time after applying Ajax, performance will increase, it may be decrease.

So, to full fill the above objective, we propose a model, which provide guidance in the form of model based testing, to the web developer that whether he/she should apply the Ajax on traditional web application or not. Our paper also describes current existing performance testing techniques which are used to measure the performance of Ajax based web applications.

Keywords: Performance Testing, Ajax, Web application Testing.

1. INTRODUCTION

In the last decade, with the prevalence of the internet, Web applications have grown quickly because of it interaction, convenient and universality. Such web applications are being use to support a wide range of internet activities like business transaction such as E-Commerce and scientific activities such as information searching and sharing. However, since web application’ carrier, the internet have characters such as distributed, dynamic, interactive, multiplatform, which make their running environment more heterogeneous and autonomous [1].

A typical web application works as follows: user supplies input to browser, browser send a request for the URL to the server, web server respond to a page- browser wait for users next input and then goes back to the first step. Key points to here are: the browser issue request to the entire page and the entire page get refreshed as a result of this action. These actions are occurring as a direct consequence of user action. These key points are the main disadvantages of traditional web application. Because every time entire page is refreshed, it takes more network utilization and until the response does not come back to the user, user remains bound with web response (i.e. cannot perform any other task with web application) [2].

In contrast with traditional web application, AJAX web applications make a number of *asynchronous* web requests for *parts* of the current web page. These requests are issued by the piece of client side code that is executed in the browser context. This client side code is usually implemented in JavaScript and is called the AJAX engine. Here an *asynchronous* request means that after sending the request user does not need to wait for response. He/ she can perform the other task with web application while response is coming from server of first request. Hence advantage of AJAX web application over the traditional web application are: since the response does not contain the entire page, a smaller amount of data gets transferred across the network thus resulting in the better network utilization, instead of reloading an entire page, AJAX applications update only part of the entire page, thus improving the responsiveness of the application [3].

Web applications develop very rapidly, but their property make it much difficult to carry out testing. Furthermore, the testing process usually needs cost, and the testing efficiency is not high. Testing AJAX application is complicated: (1) by the need of understanding the asynchronous logic, (2) by the dynamic page creation/alternation, and (3) by the bundle of technologies employed [1, 5]. A workload model specifies

statistical characterizations for a set of workload attributes that are expected to affect performance the most [7].

Objective of performance testing is to verify specified system performances (e.g. response time, service availability). It is executed by simulating hundreds or more, simultaneous user accesses over a defined time interval. The selection of the performance parameters mainly depends on the type of web site and performance requirement. Performance testing of AJAX web application is not same as the traditional web application. Two of the most widely used traditional measures for web application, “page views per unit time” and clicks per minute”, are meaningless in an AJAX context [1, 2, 5, 15]. Web performance test is mainly for the goal to maintain the performance of the system and find effective strategies of the improvement. The performance test model mainly consists of the performance parameter, performance indicators. At present, the performance of existing models uses measures mainly include throughput, response time, system resources utilization, concurrent users [6].

Our paper is organized as follows. Section 2 describes a brief description about AJAX. Section 3 give a summary based description of the related work. Section 4 describes our proposed model. The conclusion is given in the section 5.

2. AJAX

AJAX is stands for **asynchronous java script and xml** and it is a group of interrelated web development techniques used on the client side to create asynchronous web applications.

Ajax is not only a single technology but also a bundle of technology like JavaScript, CSS, HTML, and XML. With Ajax, developer can implement asynchronous communication between client and server, on the fly from data validation from data auto completion and sophisticated GUI (graphical user interface) controls based on client side component update without reloading.

Ajax (asynchronous java script and xml) support the development of rich client web application, providing primitive for the execution of asynchronous request and for the dynamic update of the page structure and the content. Often Ajax web application consist of a single page whose element are updated in response to callback activated asynchronously by the user or by a server message. Ajax allows interaction with a web server in the background. No page refresh/reloading is performed, the communication is done asynchronously via JavaScript without requiring any interaction of the user. Now combine this with the document object model, and you can create web applications that are nearly as feature rich and interactive as their desktop counterparts [4]. Benefits of AJAX are following:

Without Ajax, the client side of a web application could not communicate directly with the server without refreshing the page. Ajax makes this possible. Ajax allows the client and server of a web application to freely communicate with each other. The primary purpose of Ajax is to modify the part of the web page. As it loads only the relevant portion instead of the entire page, it makes it faster than the traditional technologies.

3. RELATED WORK

Guangzhu Jiang et al. (2009) [1] presents a testing model of web performance based on testing flow, to improve the quality and reliability of the web application. The whole process of performance testing is divided in to the five phases: planning the test, script creation, scenario definition, scenario execution and result analysis. For test case generation author divided the overall behavior of web site into the number of sub- behaviors and then generate the test case for each sub- behavior. He also clarifies that he use load Runner tool for test case generation. For performance testing author do two experiments. In first step, author determines the relationship between transaction response time, successful request rate and concurrent user of the *login transaction*. Author creates four different testing scenarios according to the quick web performance testing model. And then author find that the login transaction can meet the needs of at least 20 users requesting simultaneously, but cannot support 25 concurrent users. Second step is to determine the relationship between transaction response times, successful request rate and concurrent users of the *click transaction*. Finally author draws the conclusion that the click transaction can meet the needs of at least 50 users requesting simultaneously, but cannot support 60 concurrent users.

Manish Rajendra Dhote et al. (2012) [3] author performs a complexity analysis for performance testing on AJAX based web application. Author highlighted some complexity which may occur at the time of performance testing of AJAX based web application. The AJAX engine could make too many requests and therefore it chock up the network. Whatever criteria are chosen for performance testing, optimization of the Ajax engine is an important goal for any performance testing effort for Ajax application. Performance testing should be done not only for optimizing the AJAX engine but also to test the compatibility matrix of the browser, operating systems, client hardware, network topologies, and network speed. AJAX web application performance tester can use Firebug ti view not only script errors, but also to measure the load times of various resources such as script and image files.

Alessandro Marchetto et al. (2012) [5] gives a case study based comparison between state based testing technique (build for testing the AJAX based web application) and existing testing techniques like white box and black box, code coverage based testing. Author selected two AJAX applications: *photo share* and the *organizer*. *Photo share* manage the photo collections while the *organizer* manages an

agenda of appointments. In the experimental procedure some faults have been injected in to the original web application by a person (another than tester). These faults laid wrong and unexpected behavior. Selected web testing techniques are model based, code coverage, black box and state coverage. Each extracted test suite was applied to the faulty web applications. The result obtained for each suite is compared, with the purpose of answering our research question. According to test cases generated by other With 61 test caes in total, model based testing required less test cases than each of the other approaches, while coverage testing was most expensive. The state based approach have higher number of test cases than model based but lower than black-box testing[4].

Yasir Shoaib et al. (2011) [8] had done performance evaluation by obtaining load test measurements and by solving the LQN (Layered Queuing Network) model. Model is shown to capture the web application performance with average error of 3.77% for throughput and 12.15% for response time. For understanding model author take an example of web site namely MyBikeRoutes which represent bicycle routes in anywhere in the world. User can search best path between two points by using best path search. For a load test a think time of 7 second is added. Result shows that web application will not be able to support large number of users at satisfactory response times, showing poor performance. On the basis of functionality provided by web application, a reasonable performance objective is to sustain 40 to 50 users with a session response time of 12 seconds without think time [9]. Author build upon prior test case prioritization research and present a system-level approach to test case prioritization called Prioritization of Requirements for Test (PORT). PORT prioritizes system test cases based on four factors for each requirement: customer priority, implementation complexity, fault proneness, and requirements volatility [12].

Dian et al. (2013) [10] describe how to develop a catalog system that applies the AJAX technique and semantic technology. And new AJAX based system is compared with conventional web programming in term of page response time. In this paper author gives comparison of response time for both Ajax web application and traditional web application. Experiment shows that on the arrival of three requests, average response time is 417 ms to render the entire page. Result indicates response time of only 16 ms on average to update necessary part of the web page. This shows that AJAX has higher advantages compared with conventional or non AJAX based web pages. Theoretically, Web applications can be completely modeled with FSMs (finite state machine) as the web pages have the data, if data is there then this have the states and if states are there, they easily represented using the FSs(finite states) [11]. Asynchronous data requests (such as Ajax) can trickle data back and test-based assessment is

essential to achieve an optimal learning process for optimized the performance of web application [13, 14].

4. PROPOSED MODEL

The model works as test suite in front of web application tester. Suppose there is a web application tester, he/she has a traditional web application (non Ajax based), and he/she wants to know whether he/she should apply Ajax on that particular web application or not. Our model provides guidance to tester to fulfill his/ her aim. Existence and need of our model is valid and important because most of the time tester thinks that after applying Ajax performance will increase due to the no of advantages of Ajax, but some time it is not true i.e. performance may decrease. Consider a non AJAX based web application. We have to evaluate the performance of that web application and make sure that whether we should apply AJAX on that or not. The model consists of five steps as described below:

Step 1: Parameter calculation: In this step three parameters namely response time, think time, throughputs are calculated. Values of these parameters may be calculated by current existing formulas [6] or self generated formulas. Values of these parameters are represented by three notations respectively "RT" for response time, "TK" for think time, "TH" for throughput. For example after using formulas value of RT= 2.3 ms, TK=1.2ms, TH=12 request/min. Unit of response time and think time may be millisecond or second. And unit of throughput may be request/min or request/sec.

Step 2: Categorization of parameters in standard values: In this step we use three tables- "Table (R)" for response time. "Table (TK)" for think time, "Table (TH)" for throughput. Each Table represents set of four standards namely Very Good, Good, Average, Low. Each standard has a specific range in each table. Now value of RT, TK and TH generated by step 1 is required to fit in Table (R), Table (TK), and Table (TH) respectively. And generate a standard for each parameter. For example Table (R) is:

- Very Good: Up to 2 seconds
- Good: From 2 to 5 second
- Average: From 6 to 10 second
- Low: Over 10 second

Value of RT is 2.3 ms and if we fit this value in table then we find that it is lies between 2 to 5 second. This means that response time has a standard=good, because according to table if value RT is in between 2 to 5 second then its standard is represented by "good". And standard of remaining two parameter can be calculated same.

Step 3: Assigning value for each parameter: This step takes standard of each parameter as input. We assign a value for each parameter i.e. Very Good = 4. Good=3, Average = 2, Low=1.because the parameters are three so we get a list of

three numeric value and each numeric value will corresponding to parameter's standard. And range of standard for each parameter is 4 i.e. Very good, Good, Average, Low. Because there are number of cases like when the two parameters have same standard or all three parameter different- different standard for example case1- (Low, Low, Good), case2-(good, low, average). So for all cases, after the normalization we can get only one list like (1,1,3) which contain assigned values corresponding to standards given in case.

Step 4: *Calculation of average of all three parameter:* This step calculates average of all three parameters .i.e. sum of all three parameter is divides by 3. And this average value of all three parameters is denoted by A.

Step 5: *Criteria for apply AJAX:* In this step we proposed criteria to apply Ajax. Any organization, institute, researcher can set threshold value of A by its own. Any value of A (From 1 to 4) may be set as threshold value, based on their assumption, sampling. If values of $A \geq$ threshold value i.e. parameters are of good quality. Because A is the average of numerical values of all three parameters and the numerical value is related to parameter's standard value. Values of A will high when parameters are of good category. Hence no need to apply AJAX otherwise there is need to apply AJAX.

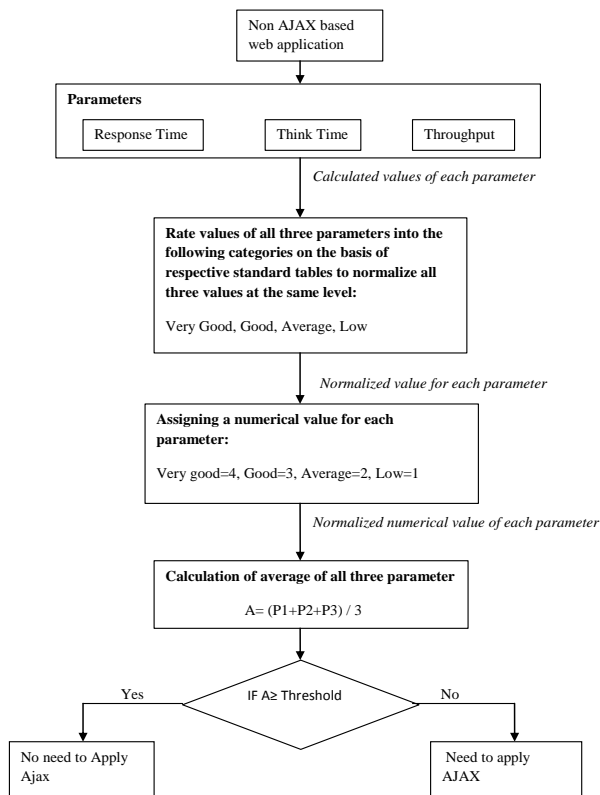


Fig 1: Model for performance testing for AJAX Based web application

5. CONCLUSIONS

The AJAX model of development for web application has rapidly gained a lot of popularity because of its promise of bringing richness and responsiveness of desktop application to the web. We propose a model which measures the performance of traditional web application in the form of quality standard value first. And then it explains that either there is a need to apply Ajax on that web application or not. Future scope in this area is that a tester can test the performance of web application by taking other more parameters like, concurrent user, CPU utilization etc.

REFERENCES

- [1]. Guangzhu Jiang, Shujuan Jiang, "A Quick Testing Model of Web Performance Based on Testing Flow And its Application", Sixth Web Information Systems And Applications Conference, Page 57-61, (2009).
- [2]. Giuseppe A. Di Lucca, Anna Rita Fasolino, "Testing Web based application: the state of the art and future trends" Elsevier Science Direct Information and software Technology 48, Page1172-1186 (2006).
- [3]. Manish Rajendra Dhote, G.G. Sarate, "Overview: Performance testing complexity analysis on AJAX model based web application" IEEE Digital object identifier 10.1109/MS (2012).
- [4]. Alessandro Marzchetto, et. al. "State based testing of AJAX web application" IRST 38050, Italy, Page 121-130 (2008)
- [5]. Alessandro Marchetto, Flippo Ricca, "A case study-based Comparison of Web testing techniques applied to Ajax Web applications" Springer, Software Tools Technical Transfer (2008).
- [6]. Zhao yang Qu, Xiaoxu, "A web performance testing model based on accessing characteristics" IPCSIT vol.25 (2012).
- [7]. Mahnaz shams, "A model based approach for testing the performance of web application" SOQUA November 6 ACM 2006.
- [8]. Yasir Shoaib, Olvisa Das, "Web application performance modeling using layered queuing networks" Electronic notes in theoretical Computer science 275,123-142 (2011).
- [9]. Jun Zuang, "optimizing Ajax web application with communication restructuring", Department of Computer Science, University of British Columbia, Canada.
- [10]. Dian Ayuba, Amirah Ismil, "Evaluation of page response time between partial and full rendering in a web based catalog system" The fourth international conference on electrical engineering and informatics (ICEEI) 2013.
- [11]. Kulvinder Singh, Rakesh Kumar, "testing web based applications using finite state machines employing genetic algorithm" International journal of engineering science and technology vol.2(12),Page 6931-6941 (2010).
- [12]. Hema Srikanth, "Improving test efficiency through system test prioritization" ELSEVIER, The Journal of Systems and Software (2012).

- [13]. Glen Oakley “Optimized Performance of Web Applications” The Elucidator, Vol. 1, No. 1, Article 1, (2006)
- [14]. Eduardo Guzmán and Ricardo Conejo, “Self-Assessment in a Feasible, Adaptive Web-Based Testing System”, IEEE Transactions on Education, vol. 48, no. 4, November 2005
- [15]. Ali Misbah, “Analysis and testing of Ajax based Single-Page Web Application”, IPA (Institute for Programming, Research and Algorithm), Dissertation Series (2009).