A NOVEL APPROACH FOR EVALUATION OF APPLYING AJAX IN THE WEB SITE

Bhupendra Singh¹, Shashank Sahu²

¹M.Tech Scholar (CSE), Ajay Kumar Garg Engineering College, Ghaziabad – UP- India ²Associate Prof. CSE Dept, Ajay Kumar Garg Engineering College, Ghaziabad – UP- India

Abstract

Now a day's mostly web applications are developing by AJAX technology due to its number of advantages over traditional web application, so instead of developing new AJAX based web application, convert the traditional web application into AJAX based web application is a good approach. It may save our time as well as our cost. But here a question can be arising is that how we can check that whether there is a need to apply AJAX on traditional web application or not. Because some time, some traditional web applications have good performance and there is no need to apply AJAX. But tester still thinks that after applying AJAX, performance of the web application will increase at each time. So to check this condition (applying AJAX or not) we develop a model.

The model works as test suite in front of web application tester. Suppose there is a web application and tester 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.

Keywords: Web application, Web application testing, AJAX, Performance testing.

1. INTRODUCTION

The internet has become a much more reliable platform for sharing information between server and client. From the classical method of providing static content on the mark-ups programming language, the worldwide technology has evolved to serve dynamic content or information retrieval that are often called web application[1].

The main issues in standard web applications are the effectiveness and efficiency in providing the information needed by the user as well as the reliability of the system to interact with the users. System response time is a another issue that requires consideration; producing quick results of the requested information, reduces user working time as well as enhance comfort of using a web-based system and experience.

In traditional web application browser issues requests for entire web pages and the entire page get refreshed as a result of this action and these requests occur as a direct consequence of user actions. The method and techniques mostly applied by system developer to solve the above described problems may be solved by using AJAX on web pages.

Ajax web application makes a number of asynchronous web requests for part of the current web page. Since the response does not contain entire page, a smaller amount of data get transfer across the network thus resulting in better network utilization and update only the part of the page, thus improving the responsiveness of the application[2]. AJAX is a combination script technique (bundle of technologies) that enables partial post back when updating the web page during a client request for information from the server. This technique improves the system response time because the browser updates only the necessary part and not the entire web page.

The main difference between traditional multi-page Web applications and AJAX-based applications is that the "more conventional" and synchronous request–response protocol has been replaced by asynchronous communications. The advantage of this change is apparent in the user interface that becomes more responsive. I.e. in traditional web applications after sending request user become bound with that web application and he cannot performs another task on that web application but in AJAX based web application after sending the request user can perform another task with that web application while response is coming back from the server.

Web application testing is difficult task. So in order to enhance the testing efficiency, appropriate testing methods and tools are needed. At present, most of the traditional testing models are confined to theoretical research. It is a pity that they are far from practical use. But in case of testing the AJAX based web application, some use full tools are present for this purpose, for example Firebug, YSlow, Tsung, Google's Load Time Analyzer. This is another advantage of AJAX web on traditional web in case of testing.

Performance testing objective 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. Information about accesses are recorded and then analyzed to estimate the load levels exhausting system resources. For Web applications, system performances is a critical issue because Web users don't like to wait too long for a response to their requests, also they expect that services are always available. Performance testing of Web applications should be considered as an everlasting activity to be carried out by analyzing data from access log files, in order to tune the system adequately. Failures uncovered by performance testing are mainly due to running environment faults (such as scarce resources, or not well deployed resources, etc.), even if any software component of the application level may contribute to inefficiency [3] [4] [5].

2. RELATED WORK

Kulvinder Singh et. al. (2010) [6] modeled web applications into Finite state machines and then applies the Genetic algorithm for generating the test cases for testing. Author describes the web application's structure in detail and then generates a state diagram for web application with the help of NFA (non deterministic finite automata). Method for test case generation is based on unique input output sequences. The UIO- method generates test from an FSM representation of the design. In addition it is assumed that IUT has the same number of states as the FSM that specifies the corresponding designs. A UIA sequence is a sequence of input and output pairs that distinguishes a state of an FSM from the remaining states. Generation of test cases uses both the test model and the test criteria. Raul Pena-Ortiz et al. (2012) [7] analyze the effect of more realistic dynamic user workload on the web performance metrics. To this author evaluates a typical e-commerce scenario and compare the result obtained using different level of dynamic workload. In this work the test suite is used to provide support to evaluate the system performance considering traditional and dynamic workload by using Dweb model. TPC-W is a transactional benchmark that models an on-line bookstore environment which is representative e-commerce system. The search group provides a book searcher by using search page to request the query and search result page to show list of results. The experimental set up used in this study is a typical two-tier configuration that consist of an Ubuntu Linux back-end and an Ubuntu Linux client from end-tier. Performance matrices are response time and total number of request per page. Author also described formulas to calculate response time and think time. Giuseppe A. Di Lucca et al. (2006) [8] presents the main difference between traditional application and web based application and how these differences impact the testing. Author also describe some relevant contribution in the field of web application testing which are developed in recent years with some indication about future trends in web application testing.

According to author need of testing is occurs due to very short time to market and market pressure, developer often neglects the testing of web application. This depreciable habit affects negatively quality of the web application and therefore triggers the need of adequate, cost effective, efficient testing approaches for verifying and validating them. The web application should be tested in order to verify its behavior when it is accessed by client. Another web application special feature to be tested is its ability to protect from unauthorized user. The main aim of testing is to discover failures in the required service/ functionality, in order to verify the conformance of the web application behavior of the specified requirement. There is so much performance testing techniques explained in literature. Each performance testing technique is beneficial in particular web application environment e.g. - model describing by Mahnaz shams et al. (2006) [9] is works when script of the system remains fix. If changes are made to a system's script than this model needs some modifications.

To perform performance testing, which parameter is the best for calculation of performance of a web site is a very important. Selection of these parameters is made according to nature of web application and their contribution in actual performance of a web site. These parameters values are change according to load on web site, server performance and network capabilities. So to select the parameter is very crucial because wrong selection of parameter may affect the overall performance of a web site. It is complex enough to find the better parameters for performance testing.

3. PROPOSED MODEL

Our proposed model is an analytical model, based on set of formulas, standard tables that provide the value of desired performance of web application and judgment of AJAX applying status. In step 1 we describe the parameters used in overall testing analysis. In step 2 we define standard table to each parameter for normalization of the parameters. In step 3 we categorize the parameters value in to their standard value. In step 4 we define a formula to calculate the overall performance of a web application. Step 5 tells us AJAX applying status. Our proposed model [13] is represented by figure 1. This model works for a single user for a web site to calculating the performance and accordingly the AJAX applying status.

3.1 Step 1: Parameter Selection and Calculation:

In our research area we select three parameters namely response time, think time, throughput. There is no restriction to choose only three parameters, if anyone wants to evaluate the performance of web application then he can take one, two or more parameters according to his simplicity in research area. Now a question can be arises is that what is the reason behind choosing only these parameters (response time, think time and throughput). Then reasons behind it are:

- Priority wise more important
- Performance mainly depends on these parameters
- Effective for our study

Another reason is that because most of the researchers use these parameters [1] [2] [7] [10] [11]. So we also take these parameters for our study.

Response time: Response time is the time when the first byte of the first http request is sent by the browser to the

server and when the last byte of the last http response is received. Unit of response time may be millisecond or second. For our experiment we take unit in seconds. Concept to generate this formula is taken from the research paper given by Raúl Peña-Ortiz et al. (2013) [7]. This formula gives the theoretical concept to calculate the response time of web application. If anyone want to calculate response time practically then he/she have to built prototype of the simple web based system in version: either AJAX enabled functions (for AJAX web application) or conventional/traditional functions (for non AJAX based web applications) or for both. Both versions will equip with the semantic technology to achieve the text search engine function support for searching the catalog information. The system prototype will be tested by using Firefox (12.0) browser and add on firebug. Then the page response time can be easily obtained. [1]. Factor on which response time is depends are: type of the transmission medium, load on the server.

Think time: Think time is the time between generating new request after evaluating response of earlier request. Unit of think time is second. Concept to generate this formula is taken from the research paper given by Raúl Peña-Ortiz et al. (2013) [7]. This formula is also gives the theoretical concept. Before calculating it practically here one thing should be clear is that think time is the time taken by the user to send new request after follow the response of earlier request. Hence it is user depended i.e. how much time he/she takes to think. So to find it practically he/she write it on note book at each time.

Throughput: Throughput is generally defines as request handled by system per unit time. Suppose at any instance response time is 5.0 second then in one minute i.e. 60 second system can handle 12 requests, means that system can generate response of 12 requests. And if response time is 10 seconds then in one minute system can handle 6 requests. Now we can see if we are increasing the response time then number of request handle per minute is decreases. Summary is that throughput is inversely proportional to response time. So we can write this as throughput = 1/response time. But in practically user does not submits the requests one by one continuously i.e. he/she must take some time(think time) for evaluation/observation to response of earlier generated request, then he/she generate the new request. So we add think time to response time. And now throughput becomes 1/ (response time + think time). There are also current existing formulas for calculation of above mentioned parameters [7] [11].

3.2 Step 2: Rate Values of all Three Parameters into the following Categories on the basis of Respective Standard Tables to Normalize all Three Values at the Same Level:

In this step we formed standard table for each parameter. Each standard table has four categories to represent the quality of parameter i.e. very good, good, average, and low. And each category has specific range of numeric values. Standard table for response time is following:

- Very Good: > 0.0 to ≤ 2.0
- Good: > 2.0 to < 5.0
- Average: ≥ 5.0 to < 10.0
- Low: ≥ 10.0 to ≤ 15

This table is made on the basis of standard table given in paper [2]. Standard table for think time is following:

- Very Good: >0.0 to ≤ 3.0
- Good: >3.0 to \leq 7.0
- Average: > 7.0 to < 8.0
- Low: ≥ 8.0 to ≤ 45.0

Diwakar Krishnamurthy et. al. (2009) [12] describe that think time of 3.0 second is low. Low in the sense of time duration, from here we find a range of think time from 0.0 to 3.0. But at that time a question can be arise is that why we put think time up to 3.0 second in very good range. For this we make a survey with peoples in the form of questionnaire. Conclusion is emerge out from survey is that most of the people suggested us that think time of up to 3.0 second should be kept in very good range. So we kept it in Very good range. Taking Think time of 3.0 to 7.0 second in good range is also based on same concept. While Raúl Peña-Ortiz et al (2013) describe that think time from greater than 7.0 but less than 8.0 is average, so we kept think time of this range in average category. Diwakar Krishnamurthy et al (2009) is also describe that think time of 45.0 second is high, high in the sense of time duration. From here we find a range from 8.0 second to 45.0 second and kept this range in low category. While standard table for throughput is following:

- Very Good: ≥ 12 to ≤ 300
- Good: ≥ 5 to ≤ 11
- Average: ≥ 3 to ≤ 4
- Low: ≥ 1 to ≤ 2

For making this table we use our proposed formula for throughput.

Throughput = 1/ (think time + response time)

Now to generate standard value of throughput we put the range of response time and think time from their standard table in above mentioned formula then we get corresponding range of throughput. For example take response time and think time range for very good.

- Very Good: >0.0 to ≤ 2.0
- Very Good: >0.0 to ≤ 3.0

Then if we put value of response time from 0.1 to 2.0 and think time from 0.1 to 3.0 in our proposed formula then we get throughput standard "Very Good: ≥ 12 to ≤ 300 " remaining standard are generated as same.

3.3 Step 3 Categorization of Parameters in

Standard Values

After step 2 we get a set of three standard values because parameters are three. In this set each parameter must has a standard value out of four standard values because range of standard for each parameter is 4 i.e. Very good, Good, Average, Low.

In first case suppose a user U1 finds a set of standard values (very good, good, average), and in case 2, User U2 find the set of values (good, average low).

Total no of possible cases= 64. For each case we have to generate a formula to calculate the performance .i.e. suppose case1 is when three parameter have standard (average, low, good) then we have to calculate the performance on the basis of this set and after using the performance formula suppose performance comes p1 and because total no cases are 64 hence performance answers will be p1, p2, p3 top64. It becomes more complex and broad and also it does not give a unique answer that performance of that web application is this. So to overcome this problem we use normalization of standard value. We simply assigned each standard value by one numeric value (Very good, good, average, low respectively by (4, 3, 2, 1). Because in standard table we arranged the standards in to descending order of performance so we also assigned them by descending order of numeric values for our simplicity. This makes easy to calculate the performance of web application to us. Now whatever the case out of 64 we have only 1 set of three assigned values which is corresponding to standard values in given case.

3.4 Step 4: Calculation of Average of all Three Parameter

We define a formula for performance evaluation.

A= (P1+P2+P3)/3

Reason behind choosing average is that because average includes factor of all three parameter.

3.5 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 Any value of A (From 1 to 4) may be set as threshold value, based on their assumption, sampling. If values of $A \ge$ 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 be high when parameters are of good categories. Hence no need to apply AJAX otherwise there is need to apply AJAX.



Fig 1 Proposed model to evaluate performance of web application

4. RESULT ANALYSIS

Result analysis is done with the help of following steps:

4.1 List the AJAX based Web Sites and Traditional

Websites

To find this we take help of web site -"Web2magazines.blog". This web site shows the list of websites those are based on AJAX. While traditional based web application are listed with the help of Google search and literature survey.

4.2 Categorize the AJAX based Web Sites on the

basis of their Popularity:

For this we take the advices from peoples, research experts. On the basis of their suggestion we listed some websites for testing. And current performance of these websites (high/low) is taken from website-"downuptime.net". Web sites taken for testing are: Facebook.com, Ebay.com, Twitter.com Youtube.com, Tagworld.com, Tmall.com, and Hao123.com.

4.3 Finding Response Time of these Web Sites:

For response time corresponding to each website we take help from website-"downuptime.net". This website shows the response time of each website in milliseconds and we converts this time in to the seconds because in overall testing process. We kept the unit of time is second.

4.4 Finding the Think Time for these Web Sites:

Because think time is user based hence we take value of think time for each web site from those users who use these website. For this we design questionnaire and took samples from the user that what should be the think time for particular web site according to them.

4.5 Throughput Calculations for these Websites:

Throughput calculation is done on the basis of a practical. Throughput is the number of requests handled by the system per unit time. So first we define the time interval in which we have to complete the practical. And perform the practical in this time duration. When time is starts we send the requests till the time is ends. And time interval between each sending request (think time) is taken from step 4. After time completion we use the formula which is following:

Number of request handled by the system = Total number of request send – number of request does not get the response.

On the basis of above mentioned formula we can calculate the throughput for each website. After finding each parameter's (response time, think time, throughput) value, take these values as input to our model which is implemented in language C. And generate outputs in term of AJAX applying status. An example for generating the output for website- "facebook.com" is following:

Parameters for facebook are: (a) Response time = 0.085 second (b) Think time = 8.0 second, (c) Throughput = 2 request/unit time.Value of response time is taken from the website "*DownUptime.Net*".Value of think time is taken from *Questionnaire* regarding to user's think time for facebook. Value of throughput is calculated by a practical. In which we set the total time for practical is 40 second. In this time duration the total nomber of request sent by us are 5 and out of 5, website does not generate response of 3 request. I.e website generate the response of only 2 request. Hense number of requests handle by system are 2 which is throughput for that web site in 40 seconds. Then our result is shows that for facebook AJAX is required.

5. MODEL VALIDITY

In this step we find out the overall validity of our model for the set of considered websites.

 Table 1 AJAX applying status and model validity for

 different websites

	Wahait	Deen	Current status	Doufoumo	0
G	websit	Kesu		Periorina	- Ou
3	es	п		nce of the	Г
r.	names	rega	according to	website	mo
N		rdin	"web2magazin	based on	del
0.		g to	e.blogspot.in"	rating	vali
		AJA	To the website	given by	dity
		Х,		"DownUp	to
		acco		Time.net"	the
		rdin		to the	mo
		g to		website	del
		our			
		mod			
		el			
1	Facebo	AJA	AJAX is using	High	100
	ok.com	X is	by this website		%
		requi			
		red			
2	Ebay.c	AJA	AJAX is using	High	100
	om	X is	by this website	U	%
		requi			
		red			
3	Twitter	AJA	AJAX is using	High	100
	.com	X is	by this website		%
		requi			
		red			
4	Youtub	AJA	AJAX is using	High	100
	e.com	X is	by this website	-	%
		requi	-		
		red			
5	Tagwor	AJA	AJAX is using	Low	100
	ld.com	X is	by this website		%
		not	-		
		requi			
		red			
6	Tmall.c	AJA	AJAX is not	High	0%
	om	X is	using by this	_	
		requi	website		
		red			
7	Hao12	AJA	AJAX is not	Low	100
	3.com	X is	using by this		%
		requi	website		
		red			

Over all validity of our model is = (100 + 100 + 100 + 100 + 100 + 100) / 7 = 85.71 %

6. CONCLUSIONS

Model evaluates performance of a web application on the basis of quality of parameters. If web application has high range parameters then it tells us that performance is good otherwise it tells that performance is not good enough and there is need to improve the performance. Judgment of parameter's quality i.e. how we can say that parameters is of good quality or low quality, it is based on parameters standards. Some standards are provided by earlier researchers and other standards are generated by us based on sampling experimental work. After normalization of standard values of parameters, we propose criteria to know whether there is need to apply Ajax on that web application or not. Any organization, institute, researcher can set threshold value for deciding the AJAX applying status. Any value of parameter A (From 1 to 4) may be set as threshold value, based on their assumption, sampling. Limitation of our model is that it is applicable for one user working. More parameters like number of concurrent users, workload and others may be taken for more accurate evaluation of the performance as a future work and a different threshold value can be set according to need.

REFRENCES

[1]. 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.

[2]. 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, IEEE, Page 57-61, (2009).

[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 Marchetto, Flippo Ricca, "A case studybased comparison of web testing techniques applied to ajax web applications" Springer Software Tools Technical Transfer (2008).

[5]. Alessandro Marzchetto, et. al. "state based testing of AJAX web application" IRST 38050, Italy, Page 121-130 (2008).

[6]. Kulvindra Singh, Rakesh kumar, " testing web based applications using finite state machines employing genetic algorithem" International journal of engineering science and technology Vol. 2(12), page-6931-6941, (2010).

[7]. Raúl Peña-Ortiz, José A. Gil, "Analyzing web server performance under dynamic user workloads", ELSEVIER journal, Computer Communications vol-36 page 386–395(2013).

[8]. 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).

[9]. Mahnaz shams, "A model based approach for testing the performance of web application", SOQUA November 6 ACM (2006).

[10]. Zhao yang Qu, Xiaoxu, "A web performance testing model based on accessing characteristics", IPCSIT vol.25 (2012).

[11]. 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).

[12]. Diwakar Krishnamurthy, Mahnaz Shams, "A Model-Based Performance Testing Toolset for Web Applications", Natural Sciences and Engineering Research Council of Canada (NSERC) and the University of Calgary, (2009). [13]. Bhupendra Singh, Shashank Sahu, "A model for performance testing of AJAX based web applications", IJRET international journal of research in engineering and technology, Volume 3, Issue 4, April 2014, pp- 889-893.

BIOGRAPHIES



Bhupendra Singh is pursuing his M. Tech degree in Computer Science & Engineering from Ajay Kumar Garg Engg. College Ghaziabad, India. He received his B.Tech. degree in Computer Science & Engineering from G. B.

Technical University (GBTU), Lucknow, India, in 2011. He is currently an active member of IEEE. His research areas are Software engineering, Networking, Operating System, Graph theory and Computer Organization.



Mr. Shashank Sahu is working as Associate Professor in Computer Science & Engineering Department at Ajay Kumar Garg Engg. College Ghaziabad, India. He received his M.Tech degree in Computer Science & Engineering from G. B. Technical University, Lucknow, India. He

is pursuing Ph.D. in Computer Science & Engineering from Sharda University, Greater Noida, India. He has 17 years of academic experience. His research areas are Software Engineering, Computer Architecture and Artificial Intelligence. He is the author of more than 10 publications in national/international journal & conferences.