International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 18 (2015) pp 38679-38683

A Novel Approach For Performance Testing On Web Application Services

Ms.B.Shyaamini,
Research Scholar, Bharathiyar University, Assistant Professor, Department of Computer Science, PSG College of Arts and Science, Coimbatore -641014 email:shyaaminibalu@gmail.com

Dr.M.Senthilkumar,
Research Supervisor, Bharathiyar University, Assistant Professor & Head, Department of Computer Science, SS.Govt Arts College, Tiruvallur-631209 email: msenthilkumar.cta.cit@gmail.com

Abstract
Nowadays, web applications are most popular in the Internet world. Most of the people spend their valuable time in web applications. Testing web applications is very essential as to ensure its quality. Performance testing measures the efficiency of web application. It identifies performance bottlenecks and optimizes system performances. The proposed approach is used to analyze web application services and to report test results. The approach uses Genetic Algorithm to generate Virtual Users which acts as workloads. The generated workloads are applied to the System Under Test (SUT) for measuring various important performance indicators. The generated test reports helps tester to measure performance of web application services under different conditions. The proposed approach is applicable for all kinds of applications like e-governance, e-commerce, e-learning, blogs, forums, social networking, and group communications.

Keywords: web applications, web applications services, performance testing, genetic algorithm, virtual user, web services

Introduction

Testing plays a vital role in software industry to ensure the Quality of the Software. Testing of software can be done in two ways namely, functional testing and non-functional testing. Functional testing purely depends on the user requirements and non-functional testing concentrates on the aspects of software performance. Performance testing validates the speed, scalability, and stability characteristics of the application. The performance testing approach are used to measure performance by means of considering various web services for web applications like e-commerce, e-learning, blogs, forums, social networking and group communications. The main focuses of testing approach are to generate workload and sending request to targeted site to measure its performance.

In this paper, we present an approach that evaluates the performance of web services for various web applications. The main contribution of this work is that the load applied to targeted site is generated using Genetic Algorithm. Here each population in genetic algorithm is treated as individual virtual user. So, boundless population is created. This leads to N number of virtual users or workloads to web application under test. It shows that the proposed approach is more flexible and scalable for performance testing on web application services.

Once workload is generated, it's applied to System under Test (Web Application) to measure various performance indexes with respect to web services. In case of any error it is send to error database. Otherwise, test result is stored in test result database. The tester analyses test reports and for measuring performance of web application's services.

This paper is organized as follows: Section 2 concentrates on literature review. The contribution of this paper is in Section 3 as proposed work with its important aspects and working scenario, features, advantages and limitations with its scopes. The work ends in Section 4 with conclusion.

Literature Review

Most of the testing experiments performed by Shea and Gallagher [1], Draheim [2], Aston [3] support creations of simple test scenarios consisting of fixed sequences of operations. With these simple test scenarios, it is hard to measure the total number of concurrent users that the system can support and the average response time for each transaction; and beside that, it is hard to identify the real bottlenecks of web applications. So, different test scenarios are to be created for handling different tests of the System Under Test (SUT). As well as, flexibility and scalability should also be considered for scripts' reusability. In order to solve these problems, a new realistic usage model with some more elaborate approaches is proposed by Xingen Wang a et al. [4] to characterize the SUT. The authors proposed two models namely, Realistic Usage Model (RUM) and Simple Load Model (SLM) to generate realistic workload for web applications. RUM analyses user's behaviors and SLM analyzes existing server logs. RUM considers only session based scenarios.

A model-based tool set for testing the performance of Web applications is given by Diwakar Krishnamurthy et al. [5]. Here, an application model that captures the inter-request dependencies and data dependencies is constructed based on inputs provided by the tester. A sequence generator uses this model to produce a large trace containing valid sequences of request types. The author defines each valid sequence of request types as a *sessionlet*. Each *sessionlet* in this trace satisfies the inter-request dependencies for the system under study. The trace of *sessionlets* is inputted to Session-Based Web Application Tester (SWAT) along with various components. It is used to obtain several sequences of requests to construct synthetic workloads with specified characteristics

The authors Draheim, D et al. [6] develops a novel approach based on stochastic model which allows an easy creation of

realistic workload of individual user behavior. Here, the user interaction with submit/ response style system is modeled as a bipartite state transition diagram. This model is used in form-oriented analysis. It is suitable for the description of user behavior. It describes what the user sees on the system's output, and what he or she provides as input to the system. In order to simulate realistic users, the author extends this model with stochastic functions that describe navigation, time delays and user input. The resulting stochastic model of user behavior can be configured in different ways as by analyzing real user data and can be used to create virtual users of different complexity. The level of detail of the stochastic model can be adjusted continuously. All this enables the load test approach to generate large sets of representative test cases. Furthermore, load test approach has its natural place in a chain of tool that support software engineers working on web applications.

Cornel Barna et al. [7] presents an adaptive performance testing model for transaction based systems. The system is modeled in two layers namely, Queuing Network Model and analytical technique. The model is self tuned and used to find the software and hardware bottlenecks in the software system. It raises an alarm as and when workload exceeds the defined threshold and saturates the given workload. During the test, the system continuously monitors the performance indicators. Model is proposed to explore the workload space and to search for points in this space that causes the worst case behavior for software and hardware components of the system. The model saturates the workload beyond the specified threshold.

Performance testing of AJAX web application is not same as the traditional web application [8]. Two of the most widely used traditional measures for web applications namely; *page views per unit time* and *clicks per minute* are meaningless in AJAX context. The performance test model mainly consists of the performance parameters and performance indicators. At present, the performance of existing models uses measures such as throughput, response time, system resources utilization, concurrent users. The authors Bhupendra Singh et al in [11] proposed an analytical model that works as a test suite for web applications and provides guidelines to convert traditional web applications into AJAX based applications. Here, the performance parameters such as think time, response time, and throughput time are considered and mean of all the three parameters are calculated. Finally, the criterion for applying AJAX is discussed. It's applicable for only one user at a time.

Fredrik Abbors et al. [9] proposed a model to generate synthetic workload, applied to the system in real-time and it measures different performance indicators. They are defined using Probabilistic Timed Automata (PTA) and they describe how different users interact with the system. It presents the evaluation of system performance. The values of different Key Performance Indicators (KPIs) of the system under a given load are identified. Number of concurrent users is measured before degradation of KPIs beyond a given threshold. The KPIs are response time, mean time between failures, number of concurrent users and throughput.

Proposed Work

In above discussed papers all the respective authors proposed various models for workload generation. Most of the models discussed above uses sophisticated scripting techniques. Here, in this paper the new concept for workload generation using Genetic Algorithm is implemented for performance testing on web application services namely HTTP, HTTPS, FTP, and SOAP with respect to various important parameters such as throughput, response time and think time. The proposed approach consists of the test plan in which test cases and script recorder are present. It is stored in test information database. Then testers setup the testing requirements such as network setup, generation of workload, services of sites to be monitored and performance indexes. The paper "Genetic Algorithm as Virtual User Manger" describes an approach where each individual in a Genetic Algorithm is treated as a Virtual user by Rasmiprava Singh et al. [10]. By expanding this idea Genetic Algorithm is used to generate virtual user in performance testing. An individual population generated by genetic algorithm is acting as virtual user, which is treated as workload for the System Under Test. So the workload is applied on System Under Test for N Number of Users request. Then test execution begins so service requests are sent to web application and service responses are received from web application. Now, measuring of performance index such as throughput, response time and think time are measured in web application services. If the targeted site is unable to reach after that, errors are stored in error database. Then the tester has to repeat test again. The successfully executed test results are stored in test result database. Then finally tester generates the report for performance testing on web application services**.**

Important aspects in proposed approach

According to our proposed approach an individual population generated by genetic algorithm is acting as virtual user, which is treated as workload for System Under Test. In Genetic Programming the individuals in an initial population are randomly generated. There are of different approaches to generating this random initial population. There are methods more widely used namely, full and grow method and other two combinations known as ramped half and half. In both methods, an initial individual are generated so that do not exceed a user specified maximum size. So population generated is the workload applied on System Under Test for N Number of Users request.

Pseudo code for full and grow methods to create initial population

procedure: gen_rnd_expr(func_set,term_set,max_d,method)
if max_d = 0 or
$\left(\text{method = grow and rand() } < \frac{|\text{termxset}|}{|\text{termxset}|+|\text{funcxset}|}\right)$
then
expr = choose_random_element(term_set)
else
func = choose_random_element(func_set)
for i = 1 to arity(func) do
arg_i = gen_rnd_expr(func_set, term_set, max_d - 1, method);

end for
expr = (func, arg_1, arg_2,...);
end if
return expr
NOTE:
func_set is a function set,
term_set is a terminal set,
max_d is the maximum allowed depth for expressions, method is either full or grow,
expr is the generated expression in prefix notation and rand() is a function that returns random numbers uniformly distributed between 0 and 1.

Advantages of Genetic Algorithm

The advantages of using genetic algorithm are given below:

- No sophisticated techniques are required
- The algorithm can able to generate N number of population according to the user request
- The algorithm is more flexible and scalable generating virtual user for performance testing on web application services.

Important Parameters in Proposed Approach Formula:

I. Average Response = sum of samples / N (unit of Measure millisecond)
II. Throughput= (N / (max (ts + t) - min (ts))) *1000 (unit of Measure millisecond)
III. Think time= Average response – response time of individual (unit of Measure millisecond)

Min = timestamp (ts) for the particular request
Max= timestamp (ts) + elapsed time (t)
N= number of samples
max (ts + t) - min(ts) = difference between max and min

The various important parameters such as throughput, response time and think time are measured using the above formula for various web application services such as HTTP, HTTPS, FTP, and SOAP.

Working scenario of Proposed Work

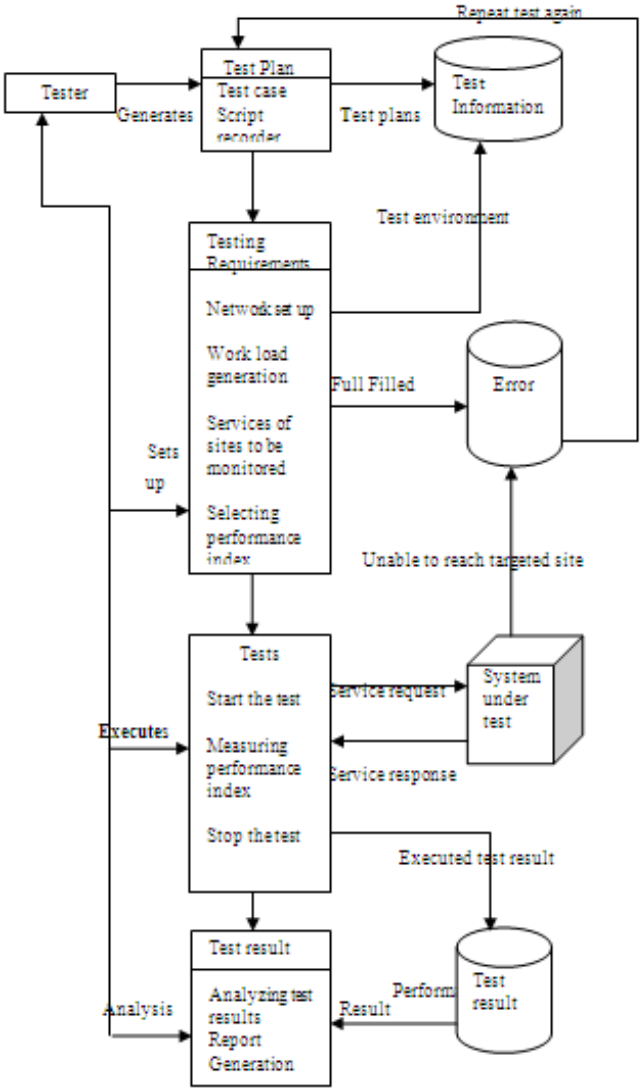

Figure 2: working scenario for a novel approach for performance testing on web application services

Features of Performance Testing Approach

The table 1 explains the various features of the newly proposed performance testing approach

Table1. Features of performance testing approach

S. No	Features	Description
1	Test Plan	➢ The test plan consists of test case and script recorder. ➢ The test case has all data about test. ➢ The various process in testing are saved using script recorder.
2	Requirements	➢ In requirements, Network set up, Work load generation, Services of sites, performance index to be monitored. ➢ Network setup plays a vital role in performance testing. ➢ The type of network used for testing has to be mentioned. ➢ Workload generation is another key aspect for performance testing. Here, Genetic Algorithm is used to generate Virtual User which acts as workload. ➢ The type of services such as HTTP, HTTPS, FTP, and SOAP are to be measured for selected web application. ➢ Performance index namely throughput, Average response time, think time etc, are to be monitored
3	Test Information	➢ The various details of test plan and requirements are stored in the test information database.
4	Test Execution	➢ In Test Execution Start the test, Request to targeted site, Measure of performance index, Stop the test are present. ➢ Once testing begins, virtual user sends request to targeted web application and performance index are measured finally testing is completed.
5	System under Test	➢ The web application for which performance index are to be measured with respect to various web services namely, HTTP, HTTPS, FTP, and SOAP
6	Error	➢ If case of any error occurs during testing it is send to error database
7	Analysis of result	➢ Once performance testing is completed analysis of test is done. It helps in understanding of the behavior for web application services under different workload conditions.
8	Test Result	➢ Test result is the database where result of testing is saved once it is completed

Advantages of Performance Testing Approach

The advantages of the proposed performance testing approach are given below.

- Each test plan is recorded using Script recorded and stored in test information database.
- The approach can able to generate virtual user by Genetic Algorithm no sophisticated techniques are required.
- The web services namely HTTP, HTTPS, FTP, and SOAP are monitored for performance measures. Most of the open source performance testing approach available now is concentrating only on HTTP, FTP, and SOAP services of web applications.
- The most important performance indexes such as throughput, Average response time, think time are measured. Most of the open source performance testing approach available now is not concentrating on think time.
- The errors during the test are stored in separate database so that we can able to check number of samples which is error during testing.
- Test result can be viewed in tables and graphs. So that it makes easy understanding for tester.
- Test results are stored in test result database so that it can be verified for future use also.
- The approach is more flexible and scalable for performance testing on web application services.

Limitations of Performance Testing Approach

The limitation of the proposed performance testing approach is given below

- The approach cannot able to measure the web services used in mobile platforms

Scope of Performance Testing Approach

The approach can enhance in such a way that it can measure performances indexes for web services used in mobile platforms. In future the number of performance indexes can be increased according to the behavior of the web application under various conditions. Here only the most important vital parameters are considered. The results of test reports can be improved by including various charts and its descriptions which makes analysis easier than now.

Conclusion

The main focus of testing approach is to generate workload and sending request to targeted site to measure its performance indexes. So we focused on an approach that evaluates the performance of web services for various web applications. The performance testing approach are used to measure performance by means of considering various web services for web applications like e-commerce, e-learning, blogs, forums, social networking and group communications. The main contribution of this work is that the load applied to targeted site is generated using Genetic Algorithm. Here each population in genetic algorithm is treated as individual virtual user. So, limitless population is created. This leads to N number of virtual users or workloads to web application under test. Once workload is generated, it's applied to System under Test to measure various performance indexes such as throughput, Average response time, think time with respect to various web services namely HTTP, HTTPS, FTP, and SOAP.

Test result is stored in test result database. The tester analyses test report and understand web application's services performance.

References:

[1]. Shea, B. and Gallagher, K., 2001. Annual load test market summary and analysis. Barnstable, MA: Newport Group Inc.

[2]. Draheim, D., Grundy, J., and Hosking, J., 2006. Realistic load testing of web applications. 10th European conference on software maintenance and reengineering, 22–24 March, Bari, Italy, 57–70.

[3]. Aston, P., 2007, "The Grinder, a Java load testing framework".

[4]. Xingen Wang a, Bo Zhou a & Wei Li a 2013 College of Computer Science and Technology, Zhejiang University, Hangzhou, Zhejiang 310007, China,Model-Based load testing of web application, Journal of the Chinese institute of engineers.

[5]. Diwakar Krishnamurthy, Mahnaz Shams, and Behrouz H. Far 2010 ―A Model-Based Performance Testing Approachset for Web Applications.

[6]. Draheim,D. Grundy, John ; Hosking, J. ; Lutteroth, C. ; Weber, G.2006 Institute of Computer Science, ― Realistic load testing of Web applications, Software Maintenance and Reengineering, 2006. Proceedings of the 10th European Conference

[7]. Cornel Barna, Marin Litoiu, Hamoun Ghanbari 2011 Department of Computer Science and Engineering York University Toronto, Canada, Model-based Performance Testing (NIER Track), ACM 978-1-4503-0445-0/11/0.2011

[8]. Bhupendra Singh1, Shashank Sahu2 2014 1M.Tech Scholar (CSE), 2Associate Prof. CSE Department, Ajay Kumar Garg Engineering College, Ghaziabad – UP- India, A Model for Performance Testing of Ajax Based Web Applications IJRET: International Journal of Research in Engineering and Technology eISSN: 2319-1163 | pISSN: 2321-7308.

[9]. Fredrik Abbors, Tanwir Ahmad, Dragos¸ Trus¸can, Ivan Porres 2012 Department of Information Technologies, Abo Akademi UniversityJoukahaisenkatu 3-5 A, 20520, Turku, Finland MBPeT: A Model-Based Performance Testing Approach, VALID 2012: The Fourth International Conference on Advances in System Testing and Validation Lifecycle.

[10]. Rasmiprava singh 1, snehalatha baede 2 and sujata khobragade 3 NIT Raipur, Genetic Algorithm as Virtual Managers: IJCSI,Vol 7 Issues,Nov 2010 ISSN 1697-0814