ISSN:1991-8178

Australian Journal of Basic and Applied Sciences

Journal home page: www.ajbasweb.com

A Survey on Web Applications Performance testing

[1]Ms.B.Shyaamini and [2]Dr.M.Senthilkumar

[1]*Research Scholar, Bharathiar University, Assistant Professor, Department of Computer Science, PSG College of Arts and Science, Coimbatore -641014*
[2]*Research Supervisor, Bharathiar University, Assistant Professor, Department of Computer Science, Government Arts College, Udhagamandalam, The Nilgiris- 643 002*

ARTICLE INFO

Article history:
Received 10 October 2015
Accepted 30 November 2015
Available online 24 December 2015

Keywords:
Models, Methods and Techniques, Parameters, Tools, Performance testing, Web Application.

ABSTRACT

Nowadays, web applications are most popular in the Internet world. Most of the people spend their valuable time in web applications namely, e-commerce, e-learning, blogs, forums, social networking, group communications and multimedia. Testing web applications is very essential as to ensure its quality. It is different from testing desktop applications. It involves various quality attributes such as performance, reliability, correctness, completeness, consistency and usability. Performance testing measures the efficiency of web application. It checks web's responses against user's requests under different load conditions. It identifies performance bottlenecks and optimizes system performances. Various aspects namely, Models, Methods and techniques, Parameters, Tools are considered with respect to measuring the performance of web applications. More often, these terminologies are used interchangeably by the people in software industry and research community. But these terminologies are having their unique identity. They cannot be used interchangeably. So, in this paper, we provide a clear-cut, concise and unambiguous focus on these terminologies along with their recent related works on various aspects that helps the research and software community to carry out a reasonable, positive and effective research and development in the direction of performance testing for web applications.

© 2015 AENSI Publisher All rights reserved.

To Cite This Article: Ms.B.Shyaamini and Dr.M.Senthilkumar., A Survey on Web Applications Performance testing. ***Aust. J. Basic & Appl. Sci.,*** *9(35): 23-29, 2015*

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. Performance is evaluated by response time, throughput, and resource-utilization J.D. Meier *et al* (2007). The performance testing can be categorized as load testing and stress testing. Load testing determines or validates performance characteristics of an application when subjected to workloads and load volumes anticipated during production operations. Stress test includes tests focused on determining performance characteristics of the System Under Test (SUT) when subjected to other stressful conditions, such as limited memory, insufficient disk space, or server failure. These tests are designed to determine under what conditions an application will fail, how it will fail, and what indicators can be monitored to warn of an impending failure. These tests are applicable for all kinds of applications. Nowadays, performance testing is the most widely used testing technique for web applications like e-commerce, e-learning, blogs, forums, social networking, group communications and multimedia. There are different Models, Methods and Techniques, Parameters and Tools available for web application performance testing. In this paper, we provide a clear-cut, concise, unambiguous focus on these terminologies and their associated survey on various aspects that helps the research and software community to carry out a reasonable, positive and effective research and development in the direction of performance testing for web applications.

This paper is organized as follows: Section 2 describes and tabulates various testing models along with the respective parameters for performance testing of various web applications. Section 3 presents the well-known methods and factors of performance testing. Section 4 classifies vital parameters for performance testing according to the type of web applications. Strength and weakness of

Corresponding Author: Ms.B.Shyaamini, Research Scholar, Bharathiar University, Assistant Professor, Department of Computer Science, PSG College of Arts and Science, Coimbatore -641014

Australian Journal of Basic and Applied Sciences, 9(35) November 2015, Pages: 23-29

open source testing tools are analyzed and compared in Section 5. Finally, Section 6 gives the conclusion and future scope of performance testing for web applications.

***2. Models**:*

Model is defined as a thing used as an example to follow or imitate. Pozin *et al*. (2011) discusses performance testing, performance characteristics or nonfunctional requirements are determined. The main difficulty in performance testing is to ensure whether the test cases and results are adequate to the client views. The client cannot able to formulate testing goals and features of system's load. Normally, the planning and execution of the load experiments takes 2 to 3 months. If the task is not formulated accurately, it leads to repeat load experiments frequently. It consumes more amounts of time and cost. So, there is the need to develop automated testing models as to improve the quality and to reduce the time and cost of performance testing of web applications. An automated testing model simulates the behavior of an application under different system configurations using mathematical approximation.

Model Based Performance Testing can be categorized and defined by Pozin *et al*. (2007) as follows:

1. Evaluation testing: Estimates the performance characteristics in a single load experiment
2. Analytical testing: Establishes the dependence of performance on computing resources in a series of load experiments.
3. Configuration testing: Configures and optimizes load characteristics of the information system.
4. Regression testing: Multiple periodic loads test under fixed conditions to detect degradation of the system

Model Based Performance Testing is carried out in six phases as given below:

1. Determination of testing goals
2. Development of test procedure
3. Preparation for testing
4. Load feed
5. Data collection
6. Interpretation and analysis of results

Most of the testing experiments performed by Shea and Gallagher (2004), Draheim (2006), Aston (2007) 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*. (2013) 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. It has limitations such as usage of simple parameters and creation of too many branches for sessions. SLM considers the performance measures such as throughput, response time and maximum number of users. Think time is also an important parameter for evaluating the performance of web applications. But SLM failed to demonstrate think-time measurement.

A model-based toolset for testing the performance of Web applications is given by Diwakar Krishnamurthy *et al*. (2010). 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. But, this model is not an automated and flexible to control over the characteristics of synthetic workloads with complex dependencies.

The authors Draheim, D *et al*. (2006) 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 tool to generate large sets of representative test cases. Furthermore, load test tool has its natural place in a chain of tools that support software engineers working on web applications. But, this model does

not calculate the time delay for sessions, if the user extends the time.

Cornel Barna *et al*.(2011) 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. Here, behavior of the model cannot be identified for more number of concurrent users. This shows the limitation of the model.

Performance testing of AJAX web application is not same as the traditional web application .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* (2014) 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. But this model is applicable for only one user.

Fredrik Abbors *et al*. (2012) 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. But, think-time is not yet incorporated in KPIs List. Even though, there is the possibility to demonstrate the think-time, model does not consider it.

Table 1 classifies various testing models along with the respective parameters for performance testing of various web applications.

Table 1: Testing Models

S.No	Proposed Model	Applicable Parameters	Type of Application	Type of Performance Testing
1	RUM and SLM Session based scenario	Concurrent user, Throughput, Response Time, Maximum number of users.	Multimedia and Communities	Analytical testing
2	Inter-Request Dependencies and data Dependencies	Response time, Think time, Throughput, CPU utilization	E-commerce	Analytical testing
3	Form-Oriented Analysis model	Average number of request, Client request rate, Number of virtual user, Average session duration, Request rate.	E-transaction	Analytical testing
4	Queuing Network Model and Analytical Technique Workload	CPU utilization, Disk utilization, Waiting time and Throughput.	E-transaction	Analytical testing
5	Model validity for web sites AJAX Based	Think time, Response time, Throughput time	AJAX Based	Evaluation testing
6	PTA- Probabilistic Timed Automata, different user behavior	Response time, Maximum users	E-transaction	Regression testing

3. Methods:

Methods are defines as the quality of being well organized and systematic in thought or action. Following a proper methodology guarantees a successful performance test project Testing methodology can be defined as identifying test phases, test targets, responsibilities in line with the overall Testing Strategy. The testing methodology also includes guidelines for how to perform testing and post deployment testing in WP3.9- Technical Core Teams (2010). The steps involved in performance testing methodologies are given as follows:

3.1 Determine Realistic User Behavior:

It is important to understand the behavior of real users on the web site, and create load test scripts that reflect the same. The kinds of activities performed by the user on website have to be determined. The workloads for performance testing have to be mentioned.

3.2 Record and Edit Scripts:

The most common user behaviors on the site are decided and each of these use cases is simulated. It goes without saying that the more realistic the set of test scripts, the more valid results are going to be obtained.

3.3 Handle Dynamic Data:

Unless site is very simple, the testers have to deal with unique data coming from each user. If each user has a unique username/password, email address, or fills in forms with some other personal data then it needs parameterize scripts to pull from Dynamic Data Sets.

3.4 Construct Test Scenarios:

To run multiple users at once, it is needed to create one or more Load Test Scenarios. Scenarios can include one or more virtual user where each of which contains some number of users executing a script. Scenarios can be configured with tons of different options, such as the total number of virtual users, instructions for ramp up/down the user load and simulation of network bandwidth.

3.5 Running the Tests:

Once a scenario has been built and configured, it is time to run the test. It is important to keep a track on the test as it is running, to see how application is performing under user load. The execution can always stop a test prematurely, twist it, and repeat again if it is needed.

3.6 Analyzing Results:

Test results are analyzed and interpreted. Again the test is carried out, if required.

Table 2: Factors for performance testing aspect

S. No	Factors	Explanation
1	Virtual User	A simulated user. This is analogous to a pretend user sitting somewhere with a web browser. Each virtual user has its own session state, cookie store, etc. A load test typically consists of many virtual users running in parallel.
2	Virtual User Group	A group of virtual users. All users within a group will run the same script with the same playback options. It can have multiple groups in order to simulate multiple behaviors simultaneously.
3	Scenario	A test scenario, made up of one or more virtual user groups. Each group within the scenario can have its own settings, such as simulated network bandwidth, randomized wait times, etc.
4	Script	A sequence of repeatable commands. Commands normally represent an HTTP request or web page (optionally with static content resources, capturing rules, and more). They can also represent wait times between pages
5	Data Set	A request or web page (optionally with static content resources, capturing rules, and more) is needed. They can also represent wait times between pages.
6	Test Run	Archived results from a test that was previously run. From a historical test run, then export a HTML report.

4. Parameters for Web Performance Testing:

Parameter is defined as a numerical or other measurable factor forming one of a set that defines a system or sets the conditions of its operation. The most significant part in the web performance testing is the selection of parameters. Performance model provides measurable standards of performance, while the standard is constituted by a series of performance indicators . The performance indicators and their description are listed below

1. **Workload** - Workload is Number of concurrent user in the particular time. The unit of measure is number
2. **Processor Usage –** It is the Amount of time processor spends executing non-idle threads. It is measured using millisecond
3. **Memory use -** Amount of physical memory available to processes on a computer is called as Memory use. Bytes is unit of measure
4. **Disk time –** It i**s** Amount of time disk is busy executing a read or write request. Millisecond is unit of measure
5. **Bandwidth –** It shows the bits per second used by a network interface. The unit of measure is bytes
6. **Memory pages/second** - The number of pages written to or read from the disk in order to resolve hard page faults. Hard page faults are when code not from the current working set is called up from elsewhere and retrieved from a disk. The unit of measure bytes.
7. **Page faults/second -** The overall rate in which fault pages are processed by the processor. This again occurs when a process requires code from outside its working set. It is measured using Bytes
8. **CPU utilization-** The average amount of CPU Utilized for an application in each second. The unit of measure millisecond.
9. **Network bytes total per second -** Rate at which bytes are sent and received on the interface including framing characters. Millisecond is unit of measure

10. **Response time –** The time from when a user enters a request until the first character of the response is received. Millisecond is unit of measure

11. **Throughput –** The rate at which computer or network receives requests per second. Millisecond is used to measure the throughput

12. **Maximum active sessions -** The maximum number of sessions that can be active at once. Maximum active sessions can be measured using number.

13. **Hit ratios -** This has to do with the number of SQL statements that are handled by cached data instead of expensive I/O operations. The **Hit ratios** can be measured using number.

14. **Hits per second -** The number of hits on a web server during each second of a load test. The Hit per second can be measured using number.

15. **Database locks -** Locking of tables and databases needs to be monitored and carefully tuned. Number is used to measure the database locks

16. **Thread counts -** An applications health can be measured by the number of threads that are running and currently active. Thread counts can be measured using number

17. **Think time-** When a user receives data from a server, the user may wait several seconds to review the data before responding. This delay is known as the think time. Millisecond is used to measure the throughput

18. **Average amount of time spent by the user on the session and site-**The time taken by the user to perform a new action on the session and site. Average amount of time spent by the user on the session and site can be measured using milliseconds

4.1 Parameters for the different types of web application:

The quality of the web application is based on the performance. It can be measured by various parameters. The web applications are categorized according to their functionalities and the vital parameters are tabulated in table [4]. It is not necessary that all the performance indicators have to be used in testing the web application performance.

Table 3: Parameters for the different types of web application

S. No	Type of web application	Important parameters							
		CU	MAS	RT	ASD	DT	CPU	TT	TP
1	E-Commerce	✓	✓	✓			✓	✓	✓
2	E-Learning	✓		✓			✓		
3	E-Transactions	✓		✓	✓	✓	✓		✓
4	Social Networking	✓		✓					✓
5	E-Governing	✓	✓	✓				✓	✓
6	Multimedia and communities	✓		✓					✓
7	Blogs/Forums	✓		✓					✓
8	Static Web Applications			✓					✓

Concurrent User-CU, Maximum Active Sessions-MAS, Response Time-RT, Average session duration-ASD, Disk time-DT, CPU utilization -CPU, Throughput –TP, Think Time-TT

5. Tools for Web Performance Testing:

Tool is defined as a device or implement used to carry out a particular function . It involves testing software applications to ensure how it performs well under their expected workload. The performance testing tools are used to identify the common problem of the web applications.

The focuses of Performance testing are

1. Speed - Determines whether the application responds quickly
2. Scalability - Determines maximum user load the software application can handle.
3. Stability - Determines if the application is stable under varying loads

Common problems faced during performance testing are:

1. Long Load time
2. Poor response time
3. Poor scalability
4. CPU utilization
5. Memory utilization
6. Network utilization
7. Operating System limitations
8. Disk usage

To measure common problems and to achieve the focuses of performance testing, various open source testing tools are available. Test tools developed by authors to measure the performance of web based application are explained as follows.

Hossain, M.S (2012), discusses web based analysis. He developed test tools to design an integrated model for underlying structure of web application. In the proposed system the performance metrics are calculated using Performance Testing Tool JMeter. Web application is built using PHP. The .net technology is used for performance testing. Performance model for web application that considers performance in each step of web process model (web helix) is implemented. Here, the author tries to integrate performance model with process model for web application. This helps for an application developer to consider performance in early phases of web application design and continues

throughout the application life cycle. With performance testing using case tool JMeter, author calculate throughput, Rate of incoming visitors and No of users at particular peak time which will help in design phase of SDLC.

Sheetal S.Patil *et al*. (2012) developed Web-based analysis and test tools. They focuses on the internal structural aspect and involves in the detail of a web application. By developing an integrated prototype tool allowing users to run the test cases to test the functionality and find faults in the web application is possible. The test cases should specify the input conditions and expected output of each test as specified. It should be generated to be extensible and allow enhancements to be made for more complex and customizable tests. Load test the newly generated test cases to test the performance of the web application and generate a test report.

The open source web application performance testing tools widely used are

5.1 OpenSTA:

Palla Vinod Kumar (2008) discusses a performance tool used by application developers for load testing is presented. It is capable of performing the heavy load test and analyses for the scripted HTTP/ HTTPS.

Features:

- The tests are performed using simple scripts, recordings and it also takes into account various results and statistics.
- During load tests, Open STA graphs resource utilization information from application servers, web servers, operating platforms and database servers that help in further analysis.

Limitations:

- Does not record with Firefox, it needs IE6, Max of 1664 users per Commander Session.

5.2 Jmeter:

It is mainly considered as a performance testing tool and it can also be integrated with the test plan. In addition to the load test plan, it allows to create a functional test plan Palla Vinod Kumar (2008).

Features:

- Capability to check the performance of the SOAP, LDAP, Message-oriented middleware (MOM) via JMS, Mail (SMTP(S), POP3(S) and IMAP(S)), MongoDB (NoSQL), and Native commands or shell scripts

Limitations:

- JMeter is not a browser as it does not perform all the actions supported by browsers.
- To be more precise, it does not execute the JavaScript present in HTML pages nor does it render the html page as a browser does.
- It has limited support for JavaScript, AJAX and complicated frameworks.
- Also the total number of threads (virtual users) generated by the test plan should be less than 300 per engine.
- One of the major limitations is that everything goes through a single console.

5.3 Grinder:

It is to perform load testing, capacity testing, functional testing, stress testing and reliability testing. Web application can be testing with help of browser by recording the script using proxy Palla Vinod Kumar (2008).

Features:

- Grinder supports HTTP Client, JEdit Syntax, Apache XMLBeans, PicoContainer, Clojure, Standards etc.
- Grinder is based on 100% pure java which can work on any platform and in any operating system

Limitations:

- It is developed for distributed test.
- The shows poor results and graphical interface

Table 4: Comparison of open source performance testing tool with their important aspects

S.No	Important Aspects	OpenSTA	JMeter	Grinder
1	Protocols	HTTP 1.0 / 1.1 / HTTPS (SSL)	HTTP 1.0 / 1.1 / HTTPS (SSL), FTP, JMS, JDBC	HTTP 1.0 / 1.1 / HTTPS (SSL), support Java API testing
2	Playback Functions	Playback facilities, but no integrated comparison functions.	No integrated comparison functionality	No integrated comparison functionality
3	Parameterization	It supports Standard function for sequential, random and pseudo-random data-file access.	Standard functions for sequential, random and pseudorandom data file access	Parameterization has to be handled in the script
4	User connection speed emulation	No inbuilt features to emulate user connection speed emulation.	No inbuilt features to emulate user connection speed emulation.	No inbuilt features to emulate user connection speed emulation.
5	Reporting and Analysis	Simple charts and	Simple charts and graphs	Test results are provided in log

		graphs sufficient for analyzing key load related statistics and resource usage monitors.	for analyzing client side metrics. Can be extended to use open source reporting tools	files.

Conclusion:

The paper gives an insight into various aspects of the web application performance testing such as Models, Methods, Parameters, and Tools with its limitations. The models used for web application performance is discussed and various methods provide guidelines for it. The vital parameters for the web testing are classified according to the web application. The open source testing tools are focused with its features and limitations. Finally, the challenges for the web application performance testing are discussed. The brief survey can help the tester to know the various aspects about the web application performance testing and also for the research scholars to understand what is important during the performance testing by the perspective of research.

REFERENCES

Meier, J.D., Carlos Farre, Prashant Bansode, Scott Barber, 2007. Performance Testing Guidance for Web Applications patterns & practices.

Pozin, B.A. and I.V. Galakhov, E.S. Leasing, Varshavskoe shosse 125, Moscow, 117405.

Russia, 2011. Models in Performance Testing ISSN 0361_7688, Programming and Computer Software, 37(1): 15–25. © Pleiades Publishing, Ltd., 2011. Original Russian Text © B.A. Pozin, I.V. Galakhov, published in Programmirovanie, 37: 1.

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

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

Aston, P., 2007. “The Grinder, a Java load testing framework”.

Xingen Wang A., A. Bo Zhou and A. Wei Li, 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.

Diwakar Krishnamurthy, Mahnaz Shams, and Behrouz H. Far, 2010. A Model-Based Performance Testing Toolset for Web Applications.

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

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

Bhupendra Singh, Shashank Sahu, 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.

Fredrik Abbors, Tanwir Ahmad, Dragos, Truscan, Ivan Porres, 2012. Department of Information Technologies, Abo Akademi UniversityJoukahaisenkatu 3-5 A, 20520, Turku,

Finland MBPeT: A Model-Based Performance Testing Tool, VALID 2012: The Fourth International Conference on Advances in System Testing and Validation Lifecycle.

WP3.9- Technical Core Teams, D3.9.2 2010-Testing Methodology, Test Plan and Tools.

Hossain, M.S., 2012. Dept. of Electr. Eng. & Comput. Sci., North South Univ., Dhaka,

Bangladesh, Performance evaluation web testing for ecommerce web sites”, Informatics, Electronics & Vision (ICIEV), International Conference.

Sheetal, S., Patil, Prof. S.D. Joshi, S.S. Dhotre, 2012. Prediction of Performance for Web Application by Performance Modeling with JMeter International Journal of Scientific & Engineering Research, 3(12): 2229-551.

Palla Vinod Kumar, 2008. Accenture Delivery Center for Technology in India.