



Comparative Analysis of Load Testing Tools Sahi And Selenium

Pooja

Student, M.Tech, UIET, MDU, Rohtak, Haryana
Poojarana130991@gmail.com

Ms. Amita

Asst. Prof, M.Tech, UIET, MDU, Rohtak, Haryana
Amita.Infotech@gmail.com

Abstract— Automated testing provides the run time valuation of software project to obtain the execution time analysis. This testing is able to identify the module level error tracking, integration tracking, load testing and the fault identification. To identify the software errors, there are some such tools and the applications. In this paper, two of the most significant load testing tools are analyzed called Sahi and Selenium. In this paper an evaluation on web-based software testing tools; API of Sahi and Selenium, Sahi OS and Selenium IDE, based on execution time, recording and playback efficiency, browser and platform compatibility, result reporting, ease of learning and cost has been presented. It will help testers to take informed decision to choose a tool according to their requirements and resources.

Keywords : Automation load testing tools, API of Sahi and Selenium, Sahi OS, Selenium IDE

I. INTRODUCTION

Software development and software quality analysis are the parallel process defined to improve the software reliability. The software measurement is required to perform the software analysis under different metrics to improve the software effectiveness. There are number of parameters to develop a software system. These parameters include the prediction and process improvement parameters. These helps to analyze the software system under theoretical, conceptual and practical aspects. Based on the earlier analysis, the real time software systems can be analyzed and predicted for the effective analysis. There are number of existing models and parameters based on which the software systems can be analyzed. These models or approaches are adopted by different organizations to deliver quality product.

There are different points of views under which the software quality analysis can be done. These parameters include the software fault analysis, software quality analysis, software reliability analysis etc. Software engineering the effective branch of software testing system used to analyze the software system under statistical analysis. There are number of methods and models available to measure the system under different aspects. Software measurement is required under three main aspects during the development process shown in figure 1.

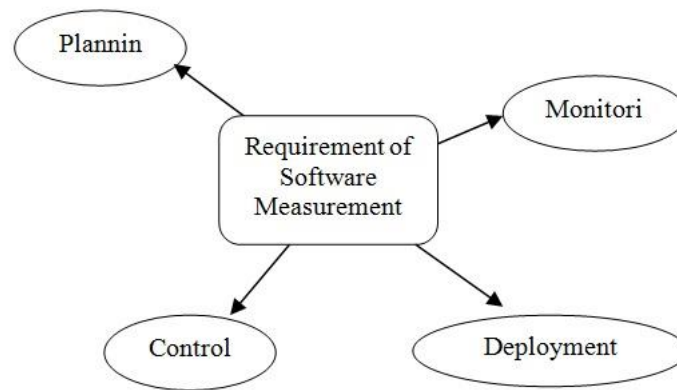


Figure 1: Software Measurement Requirements

As shown in figure, software measurement is effective to plan the software process or software system. Software plan is not a single term it defined a complete scientific process that represents the development procedure blue print. It means a software plan is able to define the software system under different aspects so that effective software development will be performed. Another aspect of software system is the monitoring of software system. Monitoring is here defined to analyze the ongoing process improvement and the objects meet till now. Another aspect of software metrics is the software control. The control is about to develop the software system under certain limits so that the software risk over the system will be minimized.

This paper is organized into five sections. Section I is the prelude of this paper, giving introduction about the topic. Section II describes literature survey. In section III web automation tools used for evaluation is discussed and section IV discusses the testing environment. In section V results discussed and section VI results and future scope and finally last section, section VII references concludes the paper.

II. RELATED WORK

In this section, some of the work provided by earlier researchers for load testing using different tools, applications and environments is presented. Author[1] has provided an analytical study on load testing for web and cloud applications as well as provided the comparative observation against traditional load testing methods. A scenario specific parameter evaluation on load testing was provided under different parameters including network traffic, delay, performance analysis criteria's. Shojae et. al.[2] has provided an effective management scenario for VMs and web services in integrated form was well as provided the evaluation for cloud based web services using load testing approach. Author has provided a work on resource driven computing method for cost evaluation respective to data storage. The managerial procedure relative to concurrency testing was provided with flexibility and operational cost analysis. Yan et. al.[3] has defined a work on web service specific load testing to derive the application based performance analysis. The defined service oriented system is here defined to provide the analysis on service load to improve the characterization of service improvement to avoid inaccurate test results. The testing specific observation is applied on different layers of cloud environment including the platform specific analysis.

Jiang et. al.[4] has provided a study on different aspects of load testing for large scale software system. The technique driven analysis under different phases is here provided to test the load and to verify the system under the behavioral observations. Author provided an evaluation for different performance measures to analyze the load and stress over the software system. Yan et. al.[5] has provided a service driven analysis on web platform to identify the criteria for load testing. The real time characterization and the performance aspect observation for requirement specific component analysis is provided. The conceptual architecture is observed by the author to generate the process comparison respective to service map so that the overall performance improvement for the application will be achieved. Wang et. al.[6] has defined a work on load testing based measure for web application specific observation so that the model driven analysis will be performed. An analytical model specification based on the variety, flexibility for evaluation under reusability modeling. The characterization under community modeling language for extending the modeling under cases. Vani et. al.[7] defined has provided a peak load specific optimal solution for web system testing. A work on infrastructure based testing load measure is provided under different QoS factors so that the requirement level analysis will be defined. A web application testing method is here provided under response time and throughput analysis. Ming et. al.[8] has defined a load testing driven load testing method for providing the phase driven load testing. A three phase method is provided by the author to formulate the load testing. At the initial phase, the load test is designed which is later on executed and analyzed for web environment. Jiang et. al.[9] has defined a work on load testing problem in automated way. Author provided the

analysis on concurrency aspect to handle the problem in real environment and with application dependency. The problem identification and solution was provided by author for large amount of data. A time driven checks are applied to identify the error by observing the application log. Author[10] has provided the load testing under scalable environment. The business operation and the relative connection analysis can be applied to validate the request and response on particular link. Author discussed the node specific behavior with impact of other node.

Nguyen *et. al*.[11] defined a verification model for control chart specific load testing in global web environment. Author defined the controlled observation in automated way so that the procedural estimation and the performance evaluation was provided in typical environment. An automated determines method was provided by the author to generate the passes specific modeling for open source projects. Author[12] provided the website based load testing in the global environment and defined the typical measure for quality of service under availability and the QiS transformed values. The opportunity driven estimation along with infrastructure traffic analysis was provided to allocate the resources significantly. Author[13] used the web application analysis and observation to identify the system reliability and to provided the optimization method. The evaluation was provided using load runner and test director tools and a scheme driven analysis on online examination project was provided. Author[14] defined an effectively methodology to enhance the process of load test execution. The complexity under different parameters is observed to generate the comparative performance observation. Author[15] has defined an automated assessment method for realistic projects under load testing. Author used the probabilistic markov chain model to provide the functional observation under profile specific mapping. The algorithm specific evaluation was provided for industrial projects under performance and fault occurrence measures.

III. RESEARCH METHODOLOGY

A large scale system such as ecommerce website which available globally are designed with fix infrastructure. These systems also support concurrent access by thousands of users. The load testing is about to analyze such capability of system and its infrastructure to handle large scale of user requests. Load testing identified the consequence of such heavy load on software system in terms of successful execution, failure, feature bug, and operational delay. Load testing is a system oriented testing which includes individual component testing and integration testing. There are number of available tools and software to perform load testing. The presented work is about to present an analytical study on two of such tools under different parameters. The test will be applied on different criteria to identify these parameters. Different loads on different configuration will be applied to perform the analysis. The testing process will first generate the load using load drivers and then apply the deterministic test execution under different aspects. The test monitoring and the successful functional execution will be observed to identify the impact of testing. The efficiency and reliability vector will be observed for these tools to verify the load impact on different software system. In this work, the observations will be performing on two testing tools called Sahi and Selenium.

3.1 Testing Framework

The presented work is here defined to test the web projects under automated load testing. The testing is here accomplished using Sahi and Selenium Tools. These tools are here used in two different ways

- Integrated API
- Application Tool

3.1.1 Integrated API

In this form, the Sahi code level implementation is taken and integrated to the java web environment. For this integration, a window based application is designed by using java swing framework. In this environment, the API of Sahi and Selenium tools are integrated in java environment. The user friendly interface is designed to accept the web project and apply the testing under Sahi and Selenium Tools. The evaluation is taken using process time and the memory consumption parameters. The results are explained in next chapter.

3.1.2 Application Tool

The separate tools for Sahi and Selenium are available to apply the Automated testing on web projects. For selenium this tool is Selenium IDE which is here used as the Mozilla pulgin. The tool accepted the web project and performs different test cases under automated testing. The results are derived in terms of successful and unsuccessful execution of test cases.

For Sahi the used tool is Sahi OS. The tool is open source and available in the form of application tool. The tool is having the integration with Mozilla browser and provided a vast set of application projects. It provides the interface to generate and accept the script. The web project testing is here analyzed by recording each of the integrated activity.

3.1.3 Sahi OS

It is the open source tool available to provide the automated testing under different aspects for web applications and services. Tool provides the user friendly environment to accept the software project as well as having a vast range of sample projects. The tool

provides the easy automation of different data specific and event specific test cases. It is able to provide the test case mapping for Complex Web 2.0 application along with AJAX interfacing.

3.3.3 Selenium IDE

Selenium IDE (Integrated Development Environment) is an automated tool that provides the web application analysis under acceptance and load testing. It is able to provide the analysis of web applications for different inputs and provides the support against different events including the mouse processing, menu processing and key board events. The tool is integrated updated within the browser as the plug-in and provides the command driven parametric analysis with context estimation. The selenium script is defined under the syntax formed analysis applied to generate the effective time saving results.

IV. RESULTS

In this present work, a graphical interface is generated to test the web applications and services under load testing. The APIs of Sahi and Selenium tools are combined here in this tool and the comparative observations are taken for the Sahi and Selenium Tools. The observation of some sample projects is also applied on open source tool of Sahi and Selenium. In this section, all the comparative results generated from the designed tool as well as from existing open source tools are described.

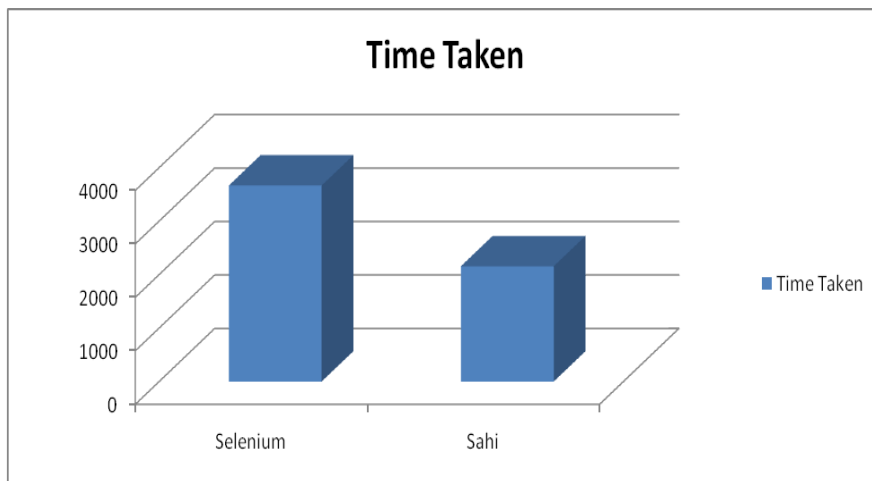


Figure 2 : Test Time Analysis

Here figure 2 is showing the comparative results obtained for the Selenium and Sahi under response time analysis is provided. The results show that the Sahi is the more effective tool to provide effective automated testing.

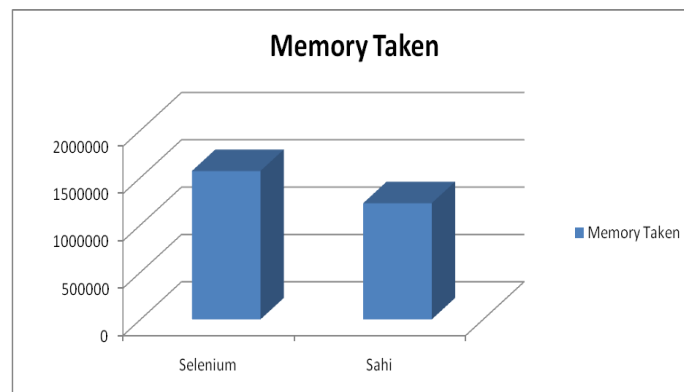


Figure 3: Memory Usage Analysis

Here figure 3 is showing the comparative results obtained for the Selenium and Sahi under memory utilization analysis is provided. The results show that the Sahi is the more memory effective tool to provide effective automated testing.

Another tool based observation relative to the successful execution of the project on Sahi OS environment are shown here in figure 4. To derive more effective conclusion, the implementation of the work is applied on three different projects. The result evaluation is shown here in figure 4.

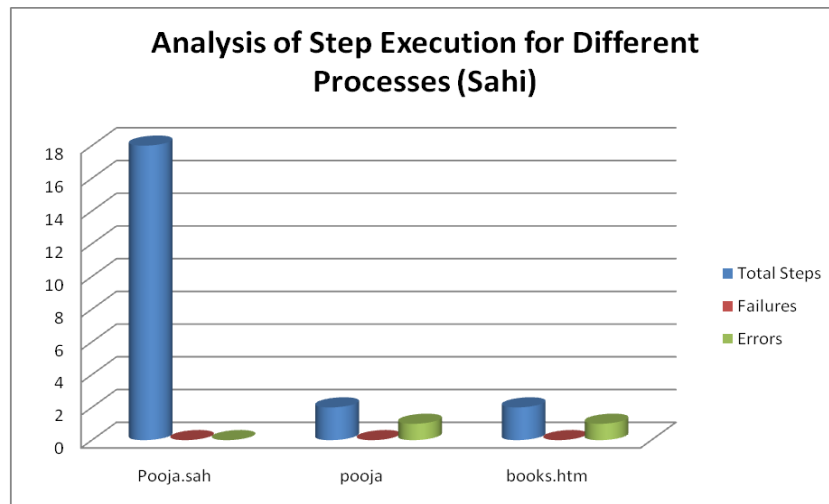


Figure 5: Analysis of Different Projects under Execution Status (Sahi)

Here figure 5 is showing the evaluation of the Automated testing applied on three different projects. The figure is showing the evaluation under three parameters called total steps, failures and the errors. The Total steps are the number of test cases implied, the failures are the system failure as some error occurred. The error shows that activity failure not the system. Figure shows that in second and third project some error in particular activity occur but no failure occur in any project.

V. CONCLUSION

The reliability of a web application, service or the utility depends on the response respective to different requests applied on that application as well as the impact of the load at particular time. Because of this, there is requirement of automated testing method to analyze the load and the associated application tasks to identify the efficiency and reliability. In this presented an analytical measure of automated testing is defined under load and activity monitoring using two tools called Selenium and Sahi. The observations taken under user defined environment and the tool specific. The results shows that the Sahi provided more efficient and memory adaptive testing on web projects.

REFERENCES

- [1] M. Arslan, U. Qamar, S. Hassan and S. Ayub, "Automatic performance analysis of cloud based load testing of web-application & its comparison with traditional load testing," Software Engineering and Service Science (ICSESS), 2015 6th IEEE International Conference on, Beijing, 2015, pp. 140-144.
- [2] A. Shojaee, N. Agheli and B. Hosseini, "Cloud-based load testing method for web services with VMs management," 2015 2nd International Conference on Knowledge-Based Engineering and Innovation (KBEI), Tehran, 2015, pp. 170-176.
- [3] M. Yan, H. Sun, X. Wang and X. Liu, "Building a TaaS Platform for Web Service Load Testing," 2012 IEEE International Conference on Cluster Computing, Beijing, 2012, pp. 576-579.
- [4] Z. M. Jiang and A. E. Hassan, "A Survey on Load Testing of Large-Scale Software Systems," in IEEE Transactions on Software Engineering, vol. 41, no. 11, pp. 1091-1118, Nov. 1 2015.
- [5] M. Yan, H. Sun, X. Wang and X. Liu, "WS-TaaS: A Testing as a Service Platform for Web Service Load Testing," Parallel and Distributed Systems (ICPADS), 2012 IEEE 18th International Conference on, Singapore, 2012, pp. 456-463.
- [6] X. Wang, B. Zhou and W. Li, "Model Based Load Testing of Web Applications," International Symposium on Parallel and Distributed Processing with Applications, Taipei, 2010, pp. 483-490.
- [7] B. Vani, R. Deepalakshmi and S. Suriya, "Web based testing — An optimal solution to handle peak load," Pattern Recognition, Informatics and Mobile Engineering (PRIME), 2013 International Conference on, Salem, 2013, pp. 5-10.

- [8] Z. M. J. Jiang, "Load Testing Large-Scale Software Systems," 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering, Florence, 2015, pp. 955-956.
- [9] Z. M. Jiang, A. E. Hassan, G. Hamann and P. Flora, "Automatic identification of load testing problems," Software Maintenance, 2008. ICSM 2008. IEEE International Conference on, Beijing, 2008, pp. 307-316.
- [10] J. A. Meira, E. C. d. Almeida, Y. Le Traon and G. Sunye, "Peer-to-Peer Load Testing," 2012 IEEE Fifth International Conference on Software Testing, Verification and Validation, Montreal, QC, 2012, pp. 642-647.
- [11] T. H. D. Nguyen, B. Adams, Z. M. Jiang, A. E. Hassan, M. Nasser and P. Flora, "Automated Verification of Load Tests Using Control Charts," 2011 18th Asia-Pacific Software Engineering Conference, Ho Chi Minh, 2011, pp. 282-289.
- [12] D. A. Menasce, "Load testing of Web sites," in IEEE Internet Computing, vol. 6, no. 4, pp. 70-74, Jul/Aug 2002.
- [13] Y. Pu and M. Xu, "Load Testing for Web Applications," 2009 First International Conference on Information Science and Engineering, Nanjing, 2009, pp. 2954-2957.
- [14] H. Malik, "A methodology to support load test analysis," 2010 ACM/IEEE 32nd International Conference on Software Engineering, Cape Town, 2010, pp. 421-424.
- [15] A. Avritzer and E. R. Weyuker, "The automatic generation of load test suites and the assessment of the resulting software," in IEEE Transactions on Software Engineering, vol. 21, no. 9, pp. 705-716, Sep 1995.