

Improving Performance of a Web Application using System Testing

Amandeep Kaur¹, Kailash Bahl²

Department of Computer Science & Engineering
Patiala Institute of Engineering & Technnology¹, Patiala¹

Abstract-Today, in the expanding era of World Wide Web, there are numerous web applications that are extensively being used by the web users. Some complex web application and bulky data offered by them often lead user to poor performance and user dissatisfaction. *System Testing* is proposed by many researchers as an effective means for tackling this problem. In this paper, we present and review existing approaches which are used to improve the performance of a web application. The survey of methodologies is based on a System Testing approach proposed here and used to enhance the throughput of an application.

Keywords – *System Testing, Performance Testing, Performance Testing tools (Apache Jmeter), System Test cases.*

I.INTRODUCTION

With the rapid increase in usage of web applications in various fields, the performance parameters of these applications are deteriorating. In order to improve the performance of these applications various system testing tools are used. System testing of software is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements after the SDLC phase. The process of performing a variety of tests on a system to explore functionality and to identify the problems. System testing is usually required before and after a system is put in place. Performance testing is the process of determining the speed or effectiveness of a computer, network, software program or device. Performance testing can verify that a system meets the specifications claimed by its developer, manufacturer or vendor. The process can compare two or more devices or programs in terms of parameters such as speed, data transfer rate, bandwidth, throughput, efficiency and reliability.

1.1 Background

Testing is the process of identifying defects, where a defect is any variance between actual and expected results/behavior. Testing is the practice of making objective judgments regarding the extent to which the system (device) meets, exceeds or fails to meet stated objectives. There are two fundamental purposes of testing: verifying procurement specifications and managing risk.

1.1.1 System testing is the first time end to end testing of application on the complete and fully integrated software product before it is launched to the market. System testing tests not only the design, but also the behavior and even the believed expectations of the customer. System testing falls under the White box testing where the testing of the internal workings or code of a software application takes place.

The goal of System testing is to verify and validate the technical, business, functional and non-functional requirements of the software. It also includes the verification & validation of software application architecture. System testing is done on staging environment that closely resembles the production environment where the final software will be deployed.

1.1.2 Performance testing is the process of determining the speed or effectiveness of a computer network, software program or device. Performance testing can also be used as a diagnostic aid in locating communications bottlenecks. Often a system will work much better if a problem is resolved at a single point or in a single component. Effective performance testing can quickly identify the nature or location of a software-related performance problem.

1.1.2.1 Need of Performance Testing:-

At the highest level, performance testing is almost always conducted to address one or more risks related to expense, opportunity costs, continuity, and/or corporate reputation. Some more specific reasons for conducting performance testing include:

• Assessing release readiness:

Enabling you to predict or estimate the performance characteristics of an application in production and evaluate whether or not to address performance concerns based on those predictions. These predictions are also valuable to the stakeholders who make decisions about whether an application is ready for release or capable of handling future growth, or whether it requires a performance improvement/hardware upgrade prior to release.

Providing data indicating the likelihood of user dissatisfaction with the performance characteristics of the system. Providing data to aid in the prediction of revenue losses or damaged brand credibility due to scalability or stability issues, or due to users being dissatisfied with application response time.

• Assessing infrastructure adequacy:

- Evaluating the adequacy of current capacity.
- Determining the acceptability of stability.
- Determining the capacity of the application's infrastructure, as well as determining the future resources required to deliver acceptable application performance.
- Comparing different system configurations to determine which works best for both the application and the business.
- Verifying that the application exhibits the desired performance characteristics, within budgeted resource utilization constraints.

• Assessing adequacy of developed software performance:

Determining the application's desired performance characteristics before and after changes to the software. Providing comparisons between the applications current and desired performance characteristics.

• Improving the efficiency of performance tuning by:

Analysing the behaviour of the application at various load levels, Identifying bottlenecks in the application, Providing information related to the speed, scalability, and stability of a product prior to production release, thus enabling you to make informed decisions about whether and when to tune the system.

1.1.3 Performance Testing tools:-

There are various performance testing tools which are widely used. Below is the brief list of most widely used performance testing tools for measuring web application performance and load stress capacity. These load testing tools will ensure application performance in peak traffic and extreme stress conditions. The list includes open source performance testing tools as well. These are:-

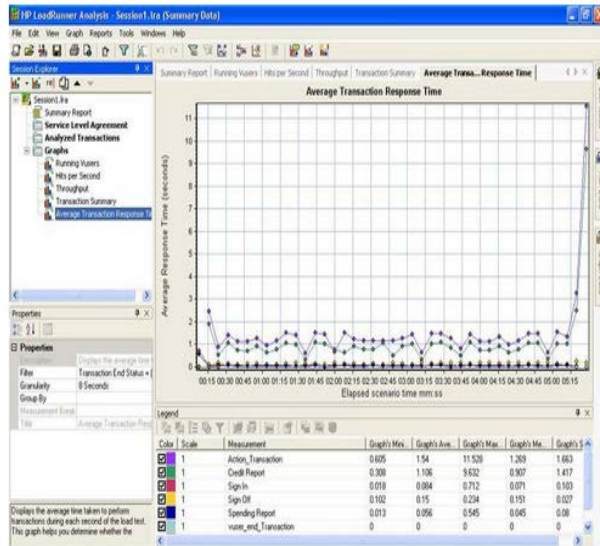
- HP loadrunner
- Apache JMeter
- Testing Anywhere
- OpenSTA

In our experiment we are using HP Load Runner 12 as the performance testing tool as it is readily available. Moreover Performance testing with HP Load Runner 12 enables difficult points to be addressed successfully. Website monitoring and load testing with HP Load Runner 12 helps slash the risks relating to poor performance. Testing the performance, scalability and reliability of web applications means our critical applications are in safer hands.

1.1.4 Test case:-

The snapshot below shows the performance enhancement after tuning the web application after making changes to the configuration parameters found out of optimal values found during performance testing.

The graph shows the performance enhancement in average transaction response time plotted against Elapsed screening time. **There is about 20 percent improvement in average transaction response time.** The test has been run on Windows XP environment using HP Load Runner 12.



III.CONCLUSION

Web applications are of much use for users as they provide a convenient and efficient interface for different services. Web applications are built, deployed and maintained to serve various functions to users with acceptable level of performance.

Sometimes the performance of the web application degrades just because of inefficient programming/unstructured techniques used. The performance of the system can be considerably improved by identifying such issues and eliminating them during performance testing.

The test case mentioned in this research paper exactly highlights this above mentioned observation regarding the performance of the web application.

REFERENCES

1. IEEE Standard ComputeMohd. Ehmer Khan “Different Approaches to White Box Testing Technique for Finding Errors” International Journal of Software Engineering and Its Applications Vol. 5 No. 3, July, 2011.
2. Graham, D., et al.: Foundations of Software Testing, Thomson, 2008, London, UK.
3. Binning, C., Kossmann, D., Lo, E.: Towards Automatic Test Database Generation, IEEE

Bulletin on Data Engineering, Vol. 31(1), 2008.

4. Martin Eisermann Performance tests with the Microsoft Internet Security and Acceleration (ISA) Server 2013, Germany.
5. Norris, Brown, Hudson, & Yoshioka, Criteria for the rating of performances, 1998, pp. 9-10.
6. Dictionary: A Compilation of IEEE Standard Computer Glossaries; IEEE; New York, NY. 1990.
7. Black, Rex; (2002). Managing the Testing Process (2nd Ed.). Wiley Publishing. ISBN 0-471-22398-0
8. Federal Highway Administration, Testing Programs for Transportation ManagementSystems – A Technical Handbook, (Washington, DC: 2007).
9. Pressman, R. (2001). Software Engineering: A Practitioner's Approach. Boston, McGrawHill.
10. "Apache JMeter - User's Manual: Building a Database Test Plan". Jakarta.apache.org. Retrieved 2013-09-20.
11. "Apache JMeter - User's Manual: Building a Web Service Test Plan". Jmeter.apache.org.
12. Xiao Jiao Yan, Xiao Wang, “Performance Testing OF Open Laboratory Management System based on Load runner”, IEEE 2011.