

Analysis of Performance Testing on Web Applications

Ms. S. Sharmila¹, Dr. E. Ramadevi²

Research Scholar, Department of Computer Science, NGM College, Pollachi, India¹

Assistant Professor, Department of Computer Science, NGM College, Pollachi, India²

Abstract: Web Applications are widely known as the building blocks of typical service oriented applications. Performance of such an application system is mainly dependent upon the components of web applications. Testing web application is nothing but to find out errors in its content, function, usability, navigability, performance, capacity, and security. Performance testing is used to determine the responsiveness, throughput, reliability, and/or scalability of a system under a given workload. This paper presents performance testing concepts, objectives, goals, types and available tools for testing web applications performance.

Keywords: Web Applications, Performance Testing, Testing Tools.

I. INTRODUCTION

“Testing is the process of executing a program with the intent of finding errors”- Glen Myers. Building a successful product hinges on two fundamental ingredients — functionality and performance. ‘Functionality’ refers to what the application lets its users accomplish, including the transactions it enables and the information it renders accessible. ‘Performance’ refers to the system’s ability to complete transactions and to furnish information rapidly and accurately despite high multi-user interaction or constrained hardware resources [2]. The performance testing involves monitoring and recording the performance levels during regular, low and high stress loads. Internet-users commonly interact with websites, many of which are dynamic in nature. These sites generate content to suit user requests instead of only serving static web pages. Due to this functionality and interactivity provided by these dynamic websites they are more appropriately considered as *Web Applications* [5].

Web testing is the name given to software testing that focuses on web applications. Complete testing of a web-based system before going live can help address issues before the system is revealed to the public. Issues such as the security of the web application, the basic functionality of the site, its accessibility to handicapped users and fully able users, as well as readiness for expected traffic and number of users and the ability to survive a massive spike in user traffic, both of which are related to load testing. The aim of Web application testing consists of executing the application using combinations of input and state to reveal failures. A failure is the manifested inability of a system or component to perform a required function within specified performance requirements [1].

I. PERFORMANCE 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 users’ 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. Core Activities of Performance Testing [6]:

- 1) Identify test environment.
- 2) Identify performance test.
- 3) Plan and design tests.
- 4) Configure test environment.
- 5) Implement test design.
- 6) Execute tests.
- 7) Analyze report and retest.

A. Performance Testing Objectives

Most of the performance tests depend on a set of predefined, documented, and agreed-upon performance objectives. Knowing the objectives from the beginning helps to make the testing process more efficient. We can evaluate our application's performance by comparing it with the performance objectives. Performance objectives often include the following:

- Response time or latency
- Throughput
- Resource utilization (CPU, network I/O, disk I/O, and memory)
- Workload

B. Goals of Performance Testing

The main goal of performance testing is to identify how well the application performs in relation to the performance objectives. Some of the other goals of performance testing include the following:

- Identify bottlenecks and their causes.
- Optimize and tune the platform configuration (both the hardware and software) for maximum performance.

- Verify the reliability of your application under stress

The following are some of the application characteristics that performance testing helps to identify:

- Response time.
- Throughput.
- Maximum concurrent users supported.
- Resource utilization in terms of the amount of CPU, RAM, network I/O, and disk I/O resources your application consumes during the test.
- Behavior under various workload patterns including normal load conditions, excessive load conditions, and conditions in between.
- Application breaking point. The application breaking point means a condition where the application stops responding to requests. Some of the symptoms of breaking point include 503 errors with a "Server Too Busy" message, and errors in the application event log that indicate that the ASPNET worker process recycled because of potential deadlocks.
- Symptoms and causes of application failure under stress conditions.
- Weak points in the application.

C. Types of Performance testing

Testing is used to verify that an application is able to perform under expected and peak load conditions, and that it can scale sufficiently to handle increased capacity. There are three types of performance tests that share similarities yet accomplish different goals:

- **Load testing:** To validate application behavior under normal and peak load conditions.
- **Stress testing:** To validate Performance the behavior when the application is pushed beyond normal load conditions.
- **Capacity testing:** To find the number of users or transactions the application can support and still meet published or promised targets.

II. PERFORMANCE TESTING WEB APPLICATIONS

Internet-users commonly interact with websites, many of which are dynamic in nature. These sites generate content to suit user requests instead of only serving static web pages. Due to this functionality and interactivity provided by these dynamic websites they are more appropriately considered as *Web Applications* [5].

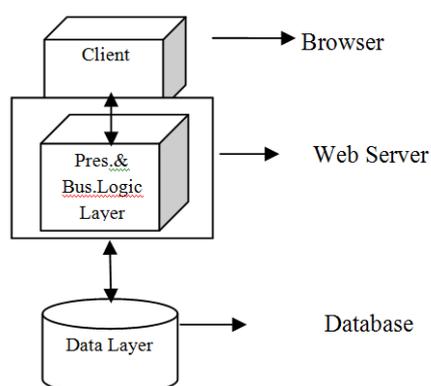


Fig. 1. Web Application Architecture

- **Client tier** (the Brower) – presents requested data.
- **Presentation tier** or **Middle Tier** or **Application Server** (the Web server) – handles all business logic and serves data to the client(s).
- **Data storage tier** (the database server) – maintains data used by the system, typically in a relational database.

The primary task of a web-application performance-testing tool is to generate load. Load is represented by number of users accessing the tested application simultaneously. Because the testing tool only simulates these users, they are called virtual users. Each virtual user represents one individual user working with the application [3].

III. TESTING TOOLS

As the software industry grows. It becomes more competitive and advanced for businesses to produce such good quality software. With this reliability and deadlines should also met. Manual testing takes too long time and it can waste the lot of time. With the aid of testing tools this can increase efficiency and get the deadlines met. Testing tool is a form of automated testing. It is basically program to do various testing tasks. Now-a-days testing is done with the help of various testing tools [7]. These are the some performance testing tools of web applications [4].

A. Apache JMeter

It is a Java platform application. It was originally designed for testing Web Applications but has since expanded to other test functions. Apache JMeter may be used to test performance both on static and dynamic resources (files, Servlets, Perl scripts, Java Objects, Data Bases and Queries, FTP Servers and more). It can be used to simulate a heavy load on a server, network or object to test its strength or to analyze overall performance under different load types.

B. NeoLoad

This is a tool used for measuring and analyzing the performance of the website. This tool analysis the performance of the web application by increasing the traffic to the website and the performance under heavy load can be determined.

C. LoadRunner

This is a HP product which can be used as a performance testing tool. It is very much useful in understanding and determining the performance and outcome of the system when there is actual load. One of the key attractive features of this testing tool is that, it can create and handle thousands of users at the same time.

D. LoadUI

LoadUI is yet another open source and load testing software used for measuring the performance of the web applications. This tool works effectively when it is integrated with the functional testing tool soapUI.

LoadUI is the most flexible and interactive testing tools. The advanced analysis and report generating features allows you to examine the actual performance by pumping in new data even while the application is being tested.

E. *WebLOAD*

WebLOAD is one such tool used for load testing and stress testing. This tool is used in the environment where there is a high demand for maximum Load testing. This tool gives the clear information on the functionality and the actual capacity of the web applications.

F. *WAPT*

WAPT refers to the Web Application Performance tool. These are scales or analyzing tools for measuring the performance and output of any web application or web related interfaces. WAPT provides detailed information about the virtual users and its output to its users during the load testing.

G. *Rational Performance Tester*

The Rational performance tester is an automated performance testing tool which can be used for a web application or a server based application where there is a process of input and output is involved.

H. *Testing Anywhere*

Testing Anywhere is a automated testing tool which can be employed for testing the performance of any web sites, web applications or any other objects. Many developers and testers make use if this tool to find out any bottlenecks in their web application and rectify them accordingly. It is a powerful tool which can test any application automatically.

I. *QEngine (ManageEngine)*

QEngine (ManageEngine) is a most common and easy-to-use automated testing tool helping in performance testing and load testing of your web applications. Many developers find it to be the most simple and easy tool to use for finding out any leakage in their web services or websites. The key important feature of this testing tool is its ability to perform remote testing of web services from any geographical location.

IV. CONCLUSION

In this paper we discussed the concepts, goals and types of performance testing for web application. In the load testing, system performance measured against the simulated user load where as in stress testing the performance of application is measured against the gradually increasing load. In the last section we described the some performance testing tools for web applications.

REFERENCES

- [1] Giuseppe A. Di Lucca, Anna Rita Fasolino, "Testing Web-based applications: The state of the art and future trends" Information and Software Technology 48, 1172–1186, 2006.
- [2] H. Sarojadevi, "Performance Testing: Methodologies and Tools", Journal of Information Engineering and Applications, ISSN 2224-5758 (print) ISSN 2224-896X (online) Vol 1, No.5, 2011.
- [3] Manjit Kaur and Raj Kumari, "Comparative Study of Automated Testing Tools: TestComplete and QuickTest Pro", International Journal of Computer Applications (0975 – 8887), Volume 24–No.1, June 2011.
- [4] Performance testing tools available at <http://www.softwaretestinghelp.com/performance-testing-tools-load-testing-tools/>
- [5] S. Elbaum, S. Karre, G. Rothermel, "Improving web application testing with user session data," in Proceedings of 25th International

Conference on Software Engineering (ICSE'03), 3-10 May 2003, pp. 49- 59.

- [6] Sheetal S.Patil and Prof. S.D.Joshi, "Identification of Performance Improving Factors for Web Application by Performance Testing", International Journal of Emerging Technology and Advanced Engineering, ISSN 2250-2459, Volume 2, Issue 8, August 2012.
- [7] Shruti N. Pardeshi, "Study of Testing Strategies and availableTools", International Journal of Scientific and Research Publications, ISSN 2250-3153, Volume 3, Issue 3, March 2013.

BIOGRAPHIES

Ms. S. Sharmila received M.C.A. degree from Bharathiar University, Coimbatore. Currently she is doing M.Phil Degree in Computer Science at Bharathiar University, Coimbatore. Her areas of interest include Data Mining, Networking, Software Testing and Quality Assurance.

Dr.E.Ramadevi received PhD degree in Computer Science from Mother Teresa Womens University, Kodaikanal. Presently she is working as an Assistant Professor in Computer science at NGM College, Pollachi, India. She has got 13 years of research experience and has more than 18 years of teaching experience. Her areas of interest include Data Mining, Knowledge base System, Intelligent and Control System and Fuzzy Logic. She has presented various papers in national and International Conferences and published 9 research papers on refereed journals.

4