

Identification of Performance Improving Factors for Web Application by Performance Testing.

Sheetal S.Patil¹, Prof. S.D.Joshi²

¹Sheetal S.Patil, M.Tech Sem IV(Computer)BVDUCOE,Pune, INDIA.

²Prof. S.D.Joshi, BVDUCOE,Pune, INDIA.

Abstract—once the coding part of any project is completed, the code must be tested to uncover maximum possible errors. Testing the 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. There are different Performance testing tools available, in this paper using Jmeter we have tried to find out factors improving performance of web application. Jmeter is a pure Java desktop application designed to load test functional behavior and measure performance.

Keywords— Performance Testing, Jmeter, load test, stress test.

I. INTRODUCTION

Almost all areas in the industry there is use of software and it grows over the years become more and more wide spread as a crucial component of many systems. System failure in any industry can be very costly. These "cost" factors call for some kind of system failure prevention. One way to ensure system's reliability is to extensively test the system. Since software is a system component, it requires a testing process. Software testing is not only detection of errors but also collection of verification, error detection and validation. The performance testing involves monitoring and recording the performance levels during regular and low and high stress loads. It tests the amount of resource usage under the just described conditions and serves as basis for making a forecast of additional resources needed (if any) in the future. It is important to note that performance objectives should have been developed during the planning stage and performance testing is to assure that these objectives are being met. However, these tests may be run in initial stages of production to compare the actual usage to the forecasted figures.

Core Activities of Performance Testing [1]:

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.

In short we do the performance testing for test the application's response, scalability, stability.

Types of performance testing:

- **Performance testing.** This type of testing determines or validates the speed, scalability, and/or stability characteristics of the system or application under test. Performance is concerned with achieving response times, throughput, and resource-utilization levels that meet the performance objectives for the project or product. In this guide, performance testing represents the superset of all of the other subcategories of performance-related testing.
- **Load testing.** This subcategory of performance testing is focused on determining or validating performance characteristics of the system or application under test when subjected to workloads and load volumes anticipated during production operations.
- **Stress testing.** This subcategory of performance testing is focused on determining or validating performance characteristics of the system or application under test when subjected to conditions beyond those anticipated during

International Journal of Emerging Technology and Advanced Engineering

Website: www.ijetae.com (ISSN 2250-2459, Volume 2, Issue 8, August 2012)

production operations. Stress tests may also include tests focused on determining or validating performance characteristics of the system or application under test 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.

Load testing

To verify application behavior under normal and peak load conditions.

- Load testing is conducted to verify that your application can meet your desired performance objectives; these performance objectives are often specified in a service level agreement (SLA). A load test enables you to measure response times, throughput rates, and resource-utilization levels, and to identify your application's breaking point, assuming that the breaking point occurs below the peak load condition [3].
- Endurance testing is a subset of load testing. An *endurance test* is a type of performance test focused on determining or validating the performance characteristics of the product under test when subjected to workload models and load volumes anticipated during production operations over an extended period of time.
- Endurance testing may be used to calculate Mean Time Between Failure (MTBF), Mean Time To Failure (MTTF), and similar metrics[6].

II. PERFORMANCE TESTING FOR WEB APPLICATION:

We will consider one scenario in which Web site is responsible for conducting online surveys with one million users in a one-hour timeframe. The site was built with wide area network (WAN) links all over the world. The site administrators want to test the site's performance to ensure that it can sustain one million user visits in one hour.

a) Performance testing of this scenario was based on following overall performance objectives

- The Web site is able to support a peak load of one million user visits in a one-hour timeframe.
- Survey submissions should not be compromised due to application errors.

b) Performance Budget/Constraints

The following budget limitations constrained the performance-testing effort:

- No server can have sustained processor utilization above 75 percent under any anticipated load (normal and peak) during submission of surveys (2 million at peak load).
- Response times for all survey submissions must not exceed 8 seconds during normal and peak loads.
- No survey submissions can be lost due to application errors.

c) Performance-Testing Objectives

The following priority objectives focused the performance testing:

- Simulate one user transaction scripted with one million total virtual users in one hour distributed among two datacenters, with half million active users at each data center.
- Simulate the peak load of one million user visits in a one-hour period.
- Test for 100-percent coverage of all survey types.
- Monitor for relevant component metrics: end-user response time, error rate, database transactions per second and overall processor, memory, and network and disk status for the database server.
- Test the error rate to determine the reliability metrics of the survey system.
- Test by using firewall and load-balancing configurations.

1) Logical Architecture:

The logical architecture used to identify the relationship between the application and the structure of the hardware and software. This information is very important when we are designing performance tests to address specific areas of concern, and when we are trying to locate a performance bottleneck.

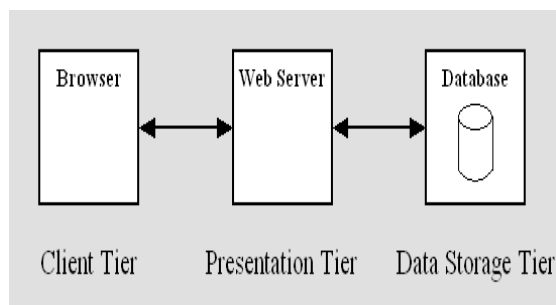


Fig.1 Three-tier Architecture [6]

- **Client tier** (the user's machine) – presents requested data.
- **Presentation tier** (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.

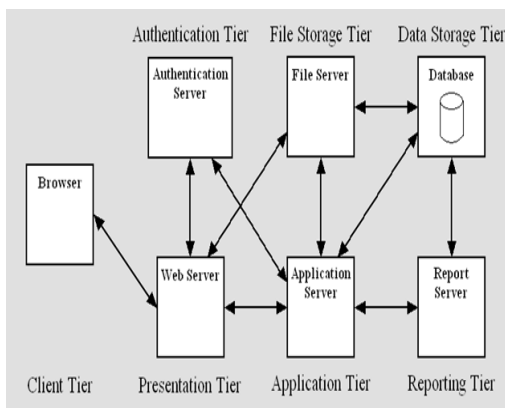


Fig 2 Multi-tier Architecture [6]

More complex architectures may include more tiers, clusters of machines that serve the same role, or even single machines serving as the host for multiple logical tiers.

2) System Architecture:

The system architecture is actually just a consolidation of the logical and physical architectures. The diagram below is an example depiction of system architecture. Obviously, it does not include every aspect of the architecture, but it does serve to highlight some points of interest for performance testing, in this case:

- Authentication and application tiers can be served by two servers.
- The mapping will allow information to better design performance tests.
- Performance tests can be targeted at the application tier directly

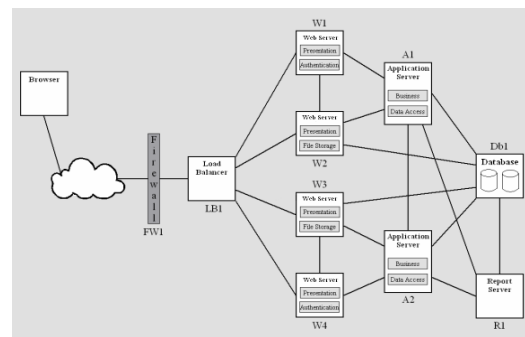


Fig 3. System Architecture [6]

III. PERFORMANCE TESTING BY JMETER

Apache JMeter is a open source and pure Java desktop application designed for performance testing [5]. 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. You can use it to make a graphical analysis of performance or to test your server/script/object behavior under heavy concurrent load.

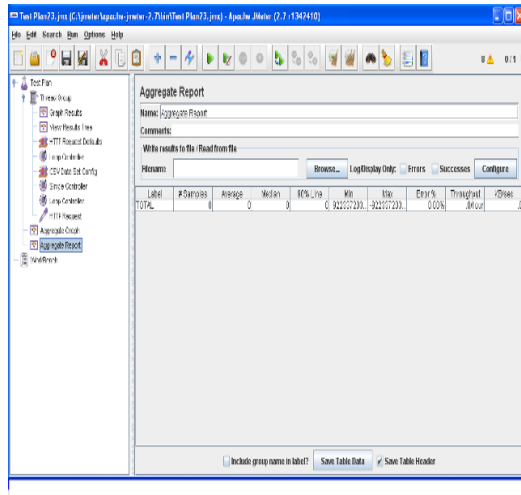


Fig4.sample test plan using Jmeter

IV. PROPOSED SYSTEM:

In this system we will be try to test web applications created on different platforms. Analysis of testing results is helpful to system designer for better choice of optimal platform to build their web applications. A performance model helps software developer that provides objective or subjective results used as input into the engineering decision cycle. The model is a mathematical representation of the characteristics of the system under consideration. Varying the parameters allows the assessment of the impact of design decisions on the performance of the system. So we try to implement performance model which will help web designer to develop a web application with improved performance.

V. CONCLUSION

In this paper we discussed the importance of performance testing for web application and various approaches in performance testing. 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. Strength testing is the longer version of load testing or stress testing. Capacity testing is the complement of load testing. In the last section we described how performance modeling helps software developer to find out bottleneck in performance of the system as well as to choose a good platform for designing web application.

REFERENCES

- [1] Kunhua Zhu Junhui Fu Yancui Li," Research the performance testing and performance improvement strategy in web application", 2nd international Conference on Education Technology and Computer, 2010, v2 328-332.
- [2] Osama HamedI and Nedat Kafri," Performance Testing for Web Based Application Architectures (.NET vs. Java EE) Performance Testing for Web Based Application Architectures(.NET vs. Java EE)" ,IEEE 2009, 978-1-4244-4615-5 ,218-224
- [3] Dr. Ramakanth Kumar P , "A Survey on Performance Testing Approaches of Web Application and Importance of WAN Simulation in Performance Testing ", International Journal on Computer Science and Engineering (IJCSE) Vol. 4 No. 05 May 2012 ISSN : 0975-3397
- [4] Scalability factors of jmeter in performance testing projects white paper
- [5] <http://jakarta.apache.org/jmeter>.
- [6] Performance Testing Guidance for Web Applications, by J.D. Meier, Carlos Farre, Prashant Bansode, Scott Barber, and Dennis Rea,Microsoft Corporation.
- [7]Daniel A. Menascé &George Mason, Load Testing, IEEE INTERNET COMPUTING JULY-august 2002 of Web Sites