

Web Application Performance Testing using LoadRunner Testing Tool

Amandeep¹, Rajkumar², Gaurav Chawla³

M.Tech Student¹, Assistant Professor & Guide², Assistant Professor³

University Institute of Engineering & Technology, MDU, Rohtak, Jamia Hamdard, New Delhi

¹amankaushik986@yahoo.com

²rajyadav76@rediffmail.com

³chawla.gaurav17@gmail.com`

Abstract: The aim of this study is the analysis and presentation of some ideas on performance testing of web application. In this paper, we made the plan of a performance testing of web application which is running on local host IIS (Internet Information Services), and got results by means of the LoadRunner which is an automatic load testing tool. We fully considered the characteristics of web application, designed the reasonable test cases, and simulated the practical scenario. In the process of running LoadRunner, we arranged the appropriate test script and test scenario, and designed the truthful test network environment. Performance testing tool LoadRunner is used to determine the responsiveness, throughput, reliability and scalability of a system under a given workload. The plan was applied to the performance testing phase of the online movie ticket booking web application. We analyzed the load testing results, proposed the improving measures, and also found the defect of the system when the massive users access the system and guided the system improvement using the test result.

Keywords: Web application, LoadRunner, Vusers, Vuser scripts.

I. INTRODUCTION

Making a web site does not end with putting all the media and software together. Actually, web site work never ends. When all the design is done, you have to test the site first before sending it to the World Wide Web for the world to see. With the development of internet technology, the web is becoming more and more important in our lives so that it has even become an essential element. At the same time, the application of the Web has never been limited to computers; it has been opened to all kinds of intelligent digital devices like mobile ones[1].

Performance is an indicator of how well a software system or component meets its requirements for timeliness. There are two important dimensions to software performance timeliness, responsiveness and scalability. Responsiveness is the ability of a system to meet its objectives for response time or throughput[2]. The response time is the time required to respond to stimuli (events). The throughput of a

system is the number of events processed in some interval of time.

Scalability is the ability of a system to continue to meet its response time or throughput objectives as the demand for the software function increases.

Any system resource such as hardware, software, or bandwidth that places defining limits on data flow or processing speed creates a bottleneck. In Web applications, bottlenecks directly affect performance and scalability by limiting the amount of data throughput or restricting the number of application connections and transactions. These problems occur at all levels of the system architecture, including the network layer, the Web server, the application server, and the database server[3].

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[4]. Due to the functionality and interactivity provided by these dynamic websites they are more appropriately considered as Web Applications.

Client/server is a program relationship in which one program (the client) requests a service or resource from another program (the server). Although the client/server model can be used by programs within a single computer, it is a more important concept for networking. In this case, the client establishes a connection to the server over a local area network (LAN) or wide-area network (WAN), such as the Internet. Once the server has fulfilled the client's request, the connection is terminated. Your Web browser is a client program that has requested a service from a server; in fact, the service and resource the server provided is the delivery of this Webpage[11].

LoadRunner works by creating virtual users who take the place of real users operating client software, such as sending requests using the HTTP protocol to IIS or Apache web servers. Requests from many virtual user clients are generated by Load Generators in order to create a load on various servers under test[8]

Software performance testing is used to determine the speed or effectiveness of a computer, network, software program or device performance testing is testing that is performed, from one perspective, to determine how fast some aspect of a system performs under a particular workload. It can also serve to validate and verify other quality attributes of the system, such as scalability, reliability and resource usage. Performance testing can serve different purposes. It can demonstrate that the system meets performance criteria[9].

A. Introduction to web application performance testing

The main aim of Web Performance Testing is to measure the actual performance of a web application and evaluate performance that the application could provide, identifying, moreover, possible bottlenecks and providing useful advice for fixing the problem (tuning of hardware components, modification of software or tuning system parameters)[6].

Before start to measure web performance effectively quite important is to know the architecture (Figure 1). This architecture includes:

- Web browser as software client where an application runs,
- Internet service providers that affect internet access
- Web servers which are applications that are able to meet requests from client (browser). Subsequently requests are forwarded to a web application which can be on the same machine or on another server,
- Application server is a place where the code of the application runs. Requests come there from web servers where are handled and provide responses,
- Database persists data of the application. Accessing the data can be difficult and because of that the access time can be too high.

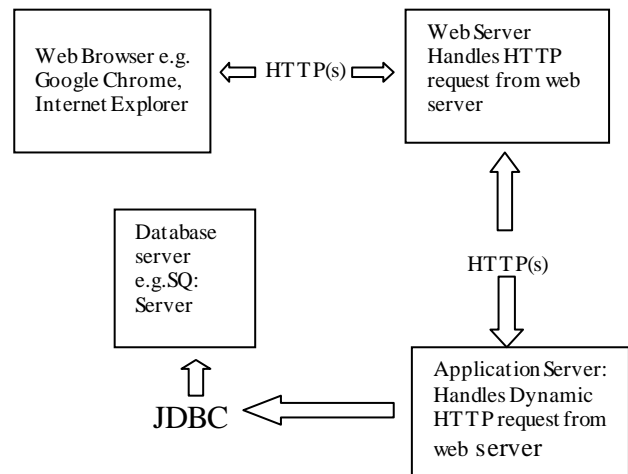


Fig 1 :Web application architecture

II. INTRODUCTION TO LOAD TESTING

Load testing is to test that if the application works fine with the loads those results from large number of simultaneous users, transactions and to determine whether it can handle peak usage periods. This paper selected the web application of online movie ticket booking as the test object, which is running on local host IIS(Internet Information Services) introduced the process of the performance testing using the LoadRunner as test tools, and found out restriction factor of the system operation based on the analysis of the test results, and then put forward the system improved scheme[8]

For example: whenever the holiday or weekends, the pages of the ticket booking sites will answer slowly, users always wait anxiously. So the performance test of the web application system before release is very important, the load testing is the important part of the performance testing. A web sites system looks perfect, and the functions can accurately be realized. However, some problems can be found by means of the load testing, such as the pressure resistance of the system and the user experience. The effective implementation of the system load testing can predict bottleneck during the congested traffic, and can solve the problem before releasing the system.

A. Importance of Load Testing

A performance-related testing process that places simulated demand on software, web applications in order to measure responses and systems behavior under both normal and anticipated peak load conditions. load testing ensures that your application can handle expected and unexpected user loads[5]. By realistically simulating user activity and monitoring infrastructure behavior, you can eliminate

most, if not all, of the bottlenecks in your web or mobile applications. The benefits of load testing are monumental. This process not only allows you to provide users with an excellent application experience, but it also helps you determine how much load an application can handle before it crashes in production, when to add another server, when to reconfigure the network, where code needs to be optimized, etc[9]. The load testing of web applications will be able to evaluate the operation of all parts of the Web server, including the CPU, memory, process, hard disk response time, etc

III PROPOSED WORK

A. Install a web application environment on local host

The web application under study, Online Movie Ticket Booking, allows the users to register and Login from client side. Users can search for the movie which are currently available in various locations and book any number of seats, after selecting the number of seats user allowed to make payment. The application uses the Visual Studio 2015 Shell is a new edition of Visual Studio offered to developers who wish to build their own web application inside Visual Studio IDE.

This application use IIS (Internet Information Services) which is an extensible web server created by Microsoft for use with windows NT family. IIS supports HTTP,HTTPS,FTP,FTPS,SMTP,NNTP and various other protocols that are used in web application communication.

Internet Information Server (IIS) is one of the most popular web servers from Microsoft that is used to host and provide Internet-based services to ASP.NET and ASP Web applications. A web server is responsible for providing a response to requests that come from users. When a request comes from client to server IIS takes that request from users and process it and send response back to users. Internet Information Server (IIS) has its own ASP.NET Process Engine to handle the ASP.NET request.

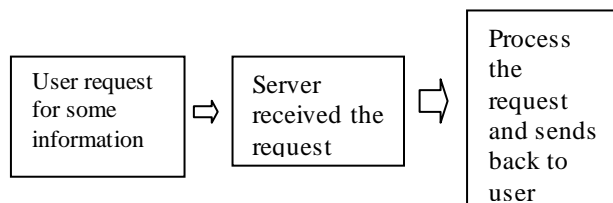


Fig 2: client server architecture

A. Load Testing Tools LoadRunner

Once your web application environment is all set, you will need to come up with a way to generate the massive amount of user traffic that web application realize. The old method of mustering up everyone in the office for a big performance test will not cut it anymore. You will need to use software to simulate the load. There are a number of tools on the market that do this. Examples would be LoadRunner and Radview's Web Load. My experience is with LoadRunner.

LoadRunner is a free open-source multi-platform tool which supports load testing of web applications and includes functionality to test the applications based on following protocols: HTTP, JDBC (for database),etc. Being open-source, enhancements can be incorporated in LoadRunner based on the specific requirements of the test, which makes it quite a useful and flexible tool. In LoadRunner, user workloads are defined by virtual user which specify the concurrent users and the requests that would be sent by virtual user.

B. The LoadRunner Testing Process

- Planning the Test
- Creating the Vuser scripts
- Creating the Scenario
- Running the Scenario
- Analyzing Test Results

Planning the test we develop a clearly defined test plan to ensure the test scenarios we develop will accomplish load-testing objectives.

Creating the Vuser scripts: we create Vuser scripts that contain tasks performed by each Vuser, tasks performed by Vusers as a whole, and tasks measured as transactions.

Creating the scenario: A scenario describes the events that occur during a testing session. It includes a list of machines, scripts, and Vusers that run during the scenario. We create scenarios using LoadRunner Controller. We can create manual scenarios as well as goal-oriented scenarios. In manual scenarios, we define the number of Vusers, the load generator machines, and percentage of Vusers to be assigned to each script. For web tests, we may create a goal-oriented scenario where we define the goal that our test has to achieve. LoadRunner automatically builds a scenario for us.

Running the Scenario: We emulate load on the server by instructing multiple Vusers to perform tasks simultaneously. Before the testing, we set the scenario configuration and scheduling. We can run the entire scenario, Vuser groups, or individual Vusers.

Analyzing Test Results: During scenario execution, LoadRunner records the performance of the application under different loads. We use LoadRunners graphs and reports to analyze the applications performance.

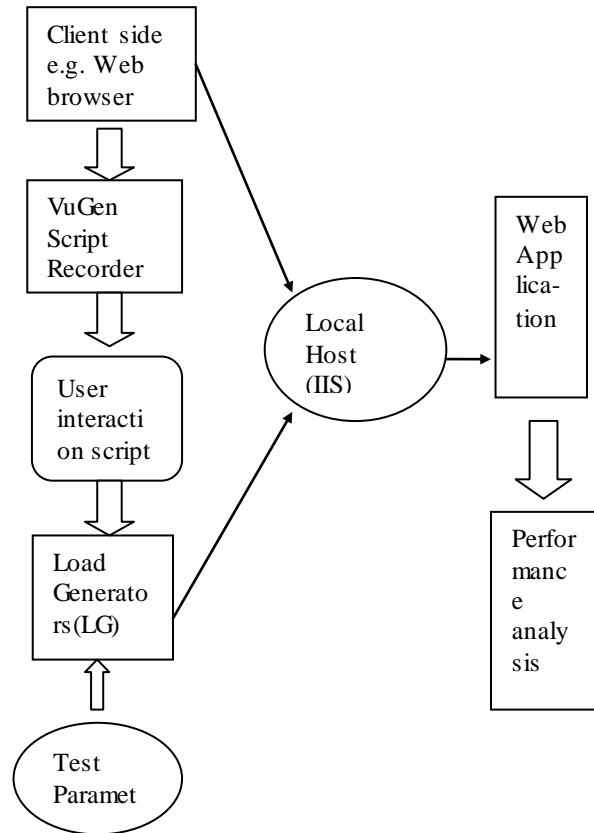


Fig 3 : Load testing process using LoadRunner

IV. RESULT AND DISCUSSION

An essential matter of this work pertains to the measurement and modeling of the web application performance. There are multiple ways how the results from the implemented test can be presented. It can be in the form of table or graphical representation. However, it is important to know, what the numbers of the table or curves in graph mean. LoadRunner can provide both forms. For this approach these analysis summary are good for checking the progress of the designed test. The load test went fine and the load test successfully executed without any

issue with respect to its throughput, number of transaction and hits per second.

A Test scenario result

In the Load testing of this Web application, we created different test scenario as below in order to Perform load testing. The test scenario were formulated keeping the overall objectives of the applications into consideration

Maximum Running Vusers:	Total Throughput (bytes):	Average Throughput (bytes/second):	Total Hits:	Average Hits per Second:	Number of Transactions Failed
2	14,259,021	70,940	1,045	5.199	0
5	37,510,516	164,520	2,750	12.061	0
10	90,013,566	276,115	6,600	20.245	0
15	139,156,990	527,110	10,260	38.864	0
20	187,522,227	429,113	13,750	31.465	0
25	255,025,004	691,125	18,700	50.678	0
30	238,507,108	792,382	17,590	58.439	0
35	313,531,686	567,992	22,990	41.649	0
40	359,999,913	631,579	26,398	46.312	2
45	386,965,081	754,318	27,081	52.789	3
50	427,451,196	791,576	791,576	58.046	5

Table 1: load testing of Login-Scenario for up to 50 users.

The above table describes the behaviour of web application in terms of total throughput , average throughput(bytes/sec), total hits ,average hits per second and the most important factor of measuring performances of web application is number of transaction failed.

The above results are helpful but they show that web application will not be able to support large number of users at satisfactory response times, showing poor performance. Based on the functionality provided by the web application, a reasonable performance objective is to sustain 40 to 50 users with a session response time of 5 seconds without think time. Considering such as objective, the web application performance has to be improved. The plan is to identify bottlenecks and remove it. The bottleneck

could be a software resource such as the Application Server, the database or it could a hardware bottleneck such as IIS, CPU.

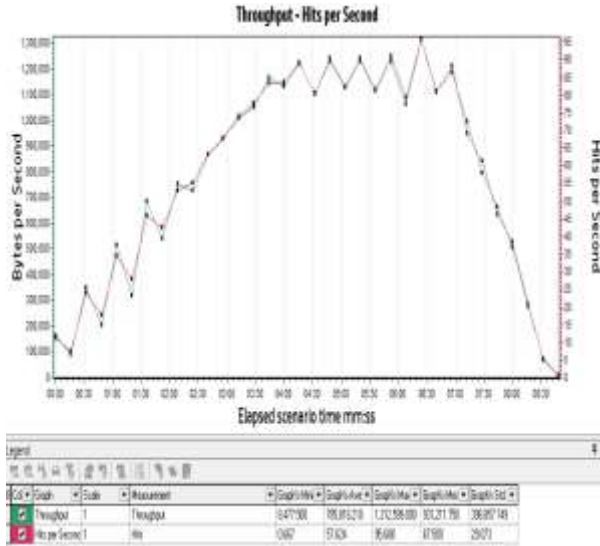


Fig 4: Throughput-Hits/second of 50 Vuser

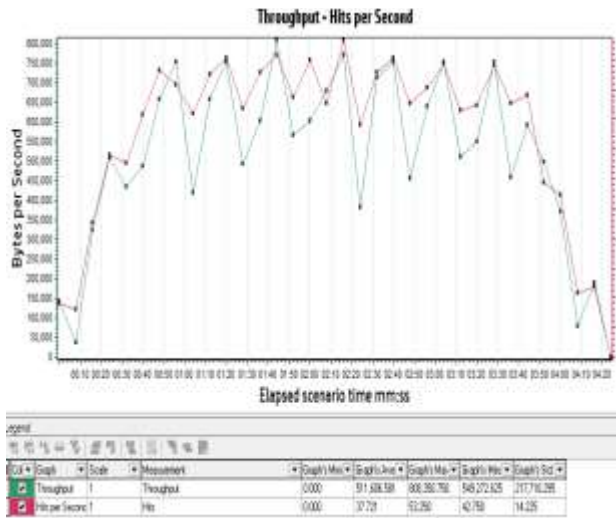


Fig 5: Throughput-Hits/second of 10 Vuser

Figure 4,5 shows the graph of Throughput-Hits/Second of 10 and 50 virtual user

Transaction Summary

Transactions: Total Passed: 558 Total Failed: 2 Total Stopped: 0 Average Response Time

	Pass	Fail	Stop
Total	558	2	0
None	558	2	0

Transaction Name	SLA Status	Minimum	Average	Maximum	Std. Deviation	90 Percent	Pass	Fail	Stop
Action Transaction	⊖	0.179	0.468	1.543	0.249	0.779	478	2	0
None		0.179	0.468	1.543	0.249	0.779	478	2	0
vuser end Transaction	⊖	0	0	0.001	0	0	40	0	0
None		0	0	0.001	0	0	40	0	0
vuser init Transaction	⊖	0.001	0.001	0.003	0.001	0.002	40	0	0
None		0.001	0.001	0.003	0.001	0.002	40	0	0

Service Level Agreement Legend: ✔ Pass ✘ Fail ⊖ No Data

Fig 6: Transactions summary of 40 Vuser

Transaction Summary

Transactions: Total Passed: 665 Total Failed: 5 Total Stopped: 0 Average Response Time

	Pass	Fail	Stop
Total	665	5	0
None	665	5	0

Transaction Name	SLA Status	Minimum	Average	Maximum	Std. Deviation	90 Percent	Pass	Fail	Stop
Action Transaction	⊖	0.182	0.537	2.393	0.296	0.939	565	5	0
None		0.182	0.537	2.393	0.3	0.939	565	5	0
vuser end Transaction	⊖	0	0	0	0	0	50	0	0
None		0	0	0	0	0	50	0	0
vuser init Transaction	⊖	0	0.007	0.168	0.028	0.004	50	0	0
None		0	0.007	0.168	0.028	0.004	50	0	0

Service Level Agreement Legend: ✔ Pass ✘ Fail ⊖ No Data

Table 7: Transactions summary of 50 Vuser

Conclusion And Future Work

As the Internet continues to expand, Web applications are becoming increasingly important sources of revenue. A literature survey revealed that users are not tolerant of delays of more than 10 seconds when using the main objective of this paper was to design and implement performance test and execute it against a web application. The test was implemented using performance testing tool LoadRunner which is a free solution for this approach. As more and more web technologies have moved a long way to create web application. Web testing plays an important role. We analyze our web application that it works fine when we gradually increase the number of virtual user but as the number of virtual user exceeds 40 to 50 the number of failed transaction increase. So we conclude that this type of testing as scale down testing that means web application working fine with 30 virtual user so if we want 3000 user can use our web application without any problem we have to scale up our server and hardware configuration hundred times.

This research work can be further enhanced by including more testing tools for comparison so that, it could find more suitable testing tools for testing the web applications. Further, some different metrics can be used for performance evaluation so that comparison could be more realistic and reliable.

References

- [1] S. Sharmila, E. Ramadevi , "Performance Evaluation And Comparison Of Web Application Testing Tools WAPT Pro and Apache JMeter", International Journal for Scientific Research Development, Vol. 3, Issue 5, August,2007
- [2] D.A. Menasce, A.F. Almeida," Challenges In Scaling E-Business Sites", Computer Measurement Group Conference, Vol. 10, Issue 15, December,2000
- [3] S. Sheetal, S.D.Joshi," Identification Of Performance Improving Factors For Web Application By Performance Testing", International Journal of Emerging Technology and Advanced Engineering, Vol.2, Issue 8, August 2012
- [4] Haroon, Shakirat, Oluwatosin , "Client- Server Model, Journal of Computer Engineering", Vol. 16, Issue 1, February 2014
- [5] Y.M. Rasal, S. Nagpure., " Web Application: Performance Testing Using Reactive Based Framework", International Journal of Research in Computer and Communication Technology, Vol. 4, Issue 2 ,February 2015
- [6] R. Khan, M. Amjad, D. Pandey , "Load Testing Of Web Applications Using Hp Load Runner", International Journal of Emerging Trends & Technology in Computer Science, Vol. 5, Issue 3, June 2016
- [7] P. Sharma," Web Application Testing (Major Challenges and Techniques)", International Journal of Modern Trends in Engineering and Research, Vol. 1, Issue 4, October - 2014
- [8] P. Fageria, M. Kaushik," Research of Load Testing and Result Based on Load Runner", International Journal of Civil Engineering Vol. 1, Issue 5, April 2014
- [9] Z. Hui-li, L. Shao-bo , " Research of Load Testing and Result Application Based on LoadRunner", National Conference on Information Technology and Computer Science, September 2008
- [10] J.K. Geetha, Monika , " Web Application Testing: A Survey", International Journal of Recent Technology and Engineering, Vol.1, Issue 6, January 2013
- [11] G. Graef , M. Gaedke," An Evolution- oriented Architecture for Web Applications", International Journal of Scientific & Engineering Research, Vol. 12, Issue 13, August 1999
- [12] R.S. Freedman, "Testability of Software Components", IEEE Transactions on Software Engineering, Vol. 17, No. 6, April 2004