Perspectives in Science (2016) 8, 355—357

Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/pisc

Performance testing (load) of web applications based on test case management☆

Rijwan Khan*, **Mohd Amjad**

Department of Computer Engineering, Jamia Millia Islamia, New Delhi, India

Received 25 January 2016; accepted 9 April 2016
Available online 29 April 2016

KEYWORDS
Software testing;
Performance testing;
Test cases;
Load test

Summary This paper includes the importance of performance testing of web applications and analyses the application's bottleneck based on hardware, software and resource utilization. The main focus of this paper is performance testing based on load test. It has been seen that everyone wants a very fast application, but at the same time, reliability of the application takes an important role; so, customer's satisfaction is the push for performance testing of a given application. Performance testing determines how fast some aspects of the system perform under a pre-defined workload. The performance testing is calculated by analyzing the production, which comes from the application hosted on the server. The main performance is calculated when business is at its peak by its hits.

Introduction

Nowadays, for every web-based application, everyone wants everything to be fast; but at the same time, there is concern about the reliability for usage. With one click on a webpage, one wants to get loaded rapidly fast so that one can finish the work soon and switch to another work. If a page takes a long time to load in the memory, one decides to give up this task. In many applications, it has been seen that people do not know about what kind of work to do and what should be the testing on the web application. So, a tester makes its own importance felt by explaining about the testing procedure for web application. The tester checks it, as the performance testing depends on the different parameters. The economic growth of any organization is affected through its web application. If a web application is not running fast, the interest of the people is lost in that business. Every organization nowadays is using the internet to conduct their online business. So, it is important to understand the performance of web application for any organization. The load testing is used to test performance of a website. This testing provides much information about the system behaviour while handling the specific load given by the customer to the system.

☆ This article belongs to the special issue on Engineering and Material Sciences.

* Corresponding author. Tel.: +91 9891676180.
E-mail address: rijwankhan786@gmail.com (R. Khan).

http://dx.doi.org/10.1016/j.pisc.2016.04.073

Table 1 Tools and utilities table.

Tools	Performance
Load Runner 11.0	The tool is used to • Capture end-user business processes and create an automated performance testing script, also known as a virtual user script. • Organize, drive, manage, and monitor the load test. • Create the load by running virtual users (VUsers). • View, dissect, and compare the performance results.
HP ALM	To schedule and run the test for client's requirements for multiple users.
Perfmon logs	To analyze the hits on the server for each request.
CA Wiley	To analyze the performance bottlenecks, if any. This would be used to monitor CPU and memory utilizations; other counters that would be monitored would be Disk utilization, process queues, JVM out of memory exceptions, etc.

The role of load testing in web application

The monitoring of the system during load testing provides very useful information about the application; the customer can take decisions and reach a proper conclusion based on these results. Load testing can also raise alarm for more complex problems that arise in the application and fix the initial problems, so that complex problems will not rise. The main purpose of the load testing is to test the application for measuring and make the report under an anticipated live load. The result during this type of testing can be end-user response time, CPU response time and memory statistics. These results give tester data to perform on it for better response on the website application (Zhu et al., 2010; Zhao and Shum, 2006; Menascé, 2002; Weyuker and Vokolos, 2000).

Table 3 Table for load test transaction load balancing.

Server	Number of requests
Application server 1	20,909
Application server 2	21,121

Test tools and utility

The following table describes the test tools and their performance (Table 1).

Experimental setup

Load run

The objective of this load test is to execute 3 h of standard test for the web application under test and determine any performance bottlenecks like high CPU utilization, memory utilization or any hardware issues. Also, failures would be also taken into consideration and would ensure that application is stable throughout the 3 h duration. In this period, spikes in response times and CPU utilization in servers have been observed (Table 2).

Observations

- Average transaction response times and 90‰ response times were under the SLA (service level agreement) of 5 s.
- All the targeted volumes for each business functionality were achieved.
- Load balancing to each server (2) was equally distributed.
- All the failures were due to data issues; but since the failures were less than 10%, it was under acceptable limits.
- CPU utilization on the application server (Tables 3—5).

Throughput

The graph in Fig. 1 highlights the amount of bytes returned by the server during the course of load test. When the users

Table 2 Table for load test and response time.

Transaction name	Average transaction response time	90% response time	Passed	Failed
Case creation submit	4.392	4.587	723	2
Increase counter submit	2.111	2.321	634	0
Reduce counter submit	2.198	2.229	650	0
Close case final submit	4.921	4.978	245	0
Display page view	3.345	3.767	378	0
View alerts info	4.290	4.332	365	1
Scan report	0.361	0.562	1	0
Manual report	0.276	0.311	1	0
Print report	0.198	0.210	1	0
Alerts report	2.453	2.489	1	0
Cases report	1.134	1.270	1	0

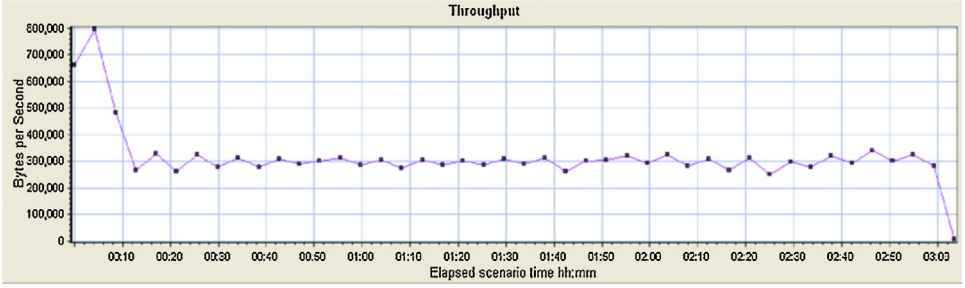

Figure 1 Load test throughput pattern.

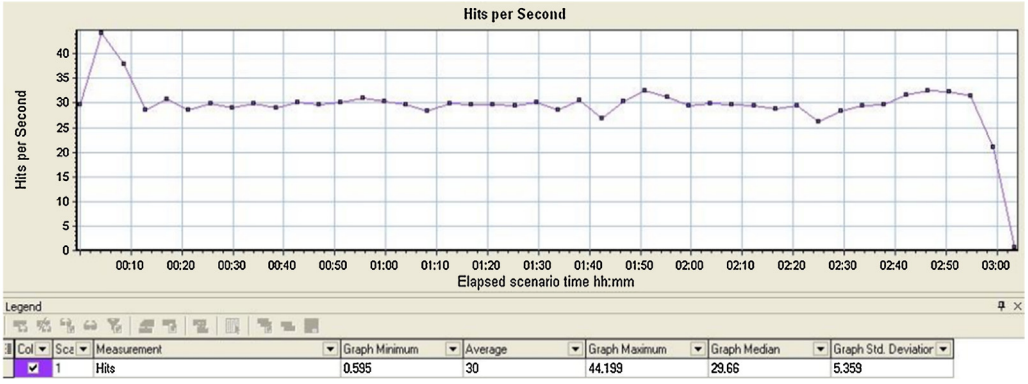

Figure 2 Load test hits per second pattern.

Table 4 Table for load test CPU utilization.

Server	Avg. CUP utilization	Max CPU utilization
Application server 1	41.60	49.01
Application server 2	43.78	48.54
Database server	65.32	82.12

Table 5 Table for load test memory utilization.

Server	Avg. CUP utilization	Max CPU utilization
Application server 1	48.31	54.45
Application server 2	55.62	67.87
Database server	50.65	72.80

were getting ramped up into the application, the amount of bytes returned by the server is highly fluctuating, and once all the users completely ramped up, then the throughput becomes stable. The below-mentioned graph shows how stable the test remained throughout, as there were no peaks or spikes during the load test period. On an average, 300,000 bytes per second were returned by the server.

Hits per second graph

The graph in Fig. 2 is for the hits per second and it shows the consistency with the throughput graph, which is a very good sign when dealing with the performance testing. The amount of hits in a second is highly fluctuating during the initial ramp-up duration, and once all the users completely ramped up, then the pattern becomes stable. On an average, 30 hits per second were observed.

Conclusion

The load has been tested with the help of HP ALM tool. Throughput and hits per second observed during the load test have been found. An application has been developed and tested on this tool, and it observed all the load tests that have been found more suitable with all other tools. Throughput and hits per second were shown to be stabilized during the load testing with HP ALM tool.

References

Menascé, D., 2002. Load testing of web sites. IEEE Internet Comput. 6 (4), 70—74.

Weyuker, E.J., Vokolos, F.I., 2000. Experience with performance testing of software systems: issues, an approach, and case study. IEEE Trans. Softw. Eng. 12, 1147—1156.

Zhao, N.Y., Shum, M.W.,2006. Technical solution to automate smoke test using rational functional tester and virtualization technology. In: Computer Software and Applications Conference, 2006, COMPSAC '06, 30th Annual International, vol. 2. IEEE.

Zhu, K., Fu, J., Li, Y., 2010. Research the performance testing and performance improvement strategy in web application. In: 2010 2nd International Conference on Education Technology and Computer, vol. 2.