

Performance Measurement of Web Applications Using Automated Tools

R. Thirumalai Selvi, N. V. Balasubramanian

Abstract—Over the last twenty years, organizations have turned to software as a means of automating work. Software applications have been used to drive huge efficiency and productivity gains as well as to provide a new medium for collaboration in a global economy. They have become the primary channel for critical information sharing. From e-mail to CRM to transaction processing, software applications are the business. Whether to deliver competitive advantage or to respond to changes in business conditions, software applications are amended on a continuous basis. Software development methodologies have matured tremendously in this time period, and the complexity of modern applications has exploded. The number of potential points of failure in a business process has correspondingly increased. In this context, rigorous performance testing has become a common strategy to both quantify and reduce this risk to a business.

Index Terms—CRM, Transaction processing, Performance Testing

I. INTRODUCTION

THE great majority of applications, especially online apps, focus on a joint and simultaneous operation of multiple clients. It is therefore necessary to use performance testing under load. Depending on the application architecture, the developer along with the customer selects the necessary tools to define the required level of performance and also the threshold number of concurrent users. Based on these data, the tester simulates the required number of concurrent users and records the quality and speed of service under such a load. Performance testing is considered as one of the most technically complex types of software testing because it requires testers with extensive knowledge and experience in both software development and the domain.

R. Thirumalai Selvi is a PhD student in Computer Science, Mother Teresa University for Women, Kodaikkanal, Tamil Nadu, India

N. V. Balasubramanian is a Professor of Computer Science & Engineering, RMK Engineering College, Kavaraipettai, Tamil Nadu, India (laks.balu@gmail.com)

II. PERFORMANCE TESTING WITH LOADRUNNER (MERCURY)

With Load Runner there are five phases listed below:

Plan Load Test: Define your performance testing requirements, for example, the number of concurrent users, typical business processes and the required response time.

Create Vuser Scripts: Capture the end-user activities into automated scripts.

Define a Scenario: Use the LoadRunner Controller to set up the load test.

Run a Scenario: Drive, manage, and monitor the load test from the LoadRunner.

Analyze the Results: Use LoadRunner Analysis tool to create graphs and reports, and evaluate performance.

TABLE I
TRANSACTION RESPONSE TIMES (SINGLE USER)

Measurement	Min.	Ave.	Max.
Action Transaction	41.276	41.276	41.276
vuser end Transaction	4.664	4.664	4.664
vuser init Transaction	4.297	4.297	4.297

FIGURE I
HITS PER SECOND (SINGLE USER)

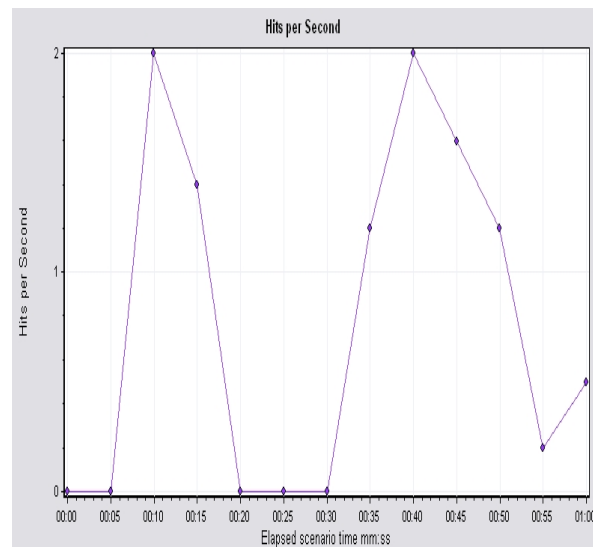


FIGURE II
HITS PER SECOND (FIVE USERS)

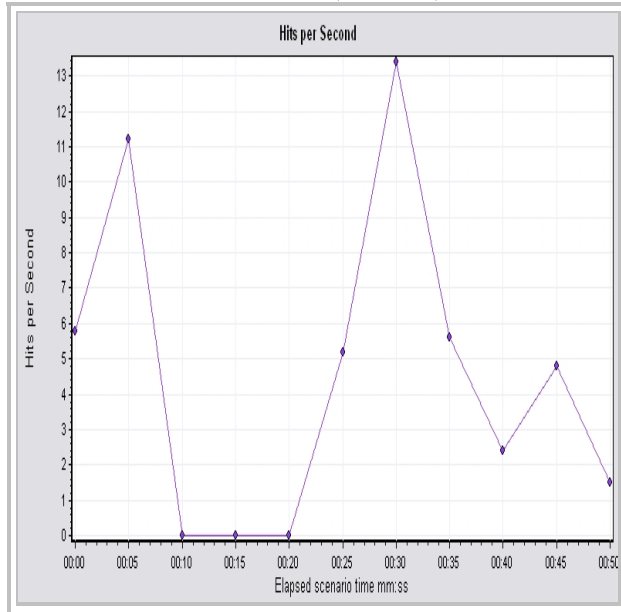


FIGURE IV
THROUGHPUT (FIVE USERS)

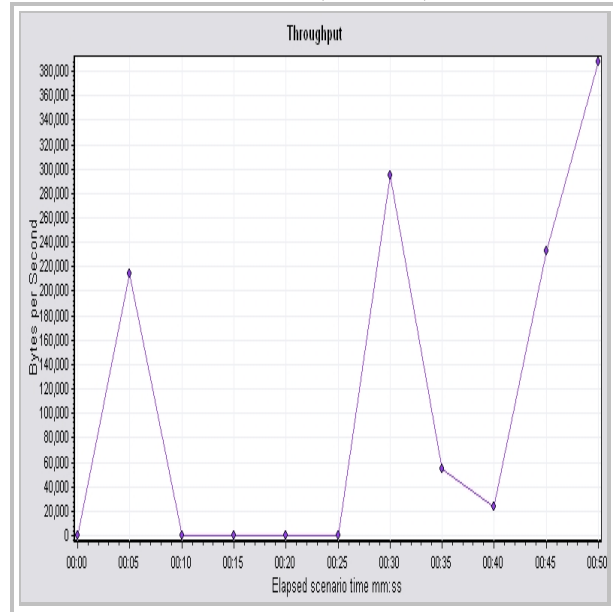


TABLE II
HITS PER SECOND (SINGLE USER)

Measurement	Min.	Ave.	Graph Max.	Graph Medium	Graph SD
Hits	0.0	0.79	2	0.5	0.777

TABLE V
THROUGHPUT (FIVE USERS)

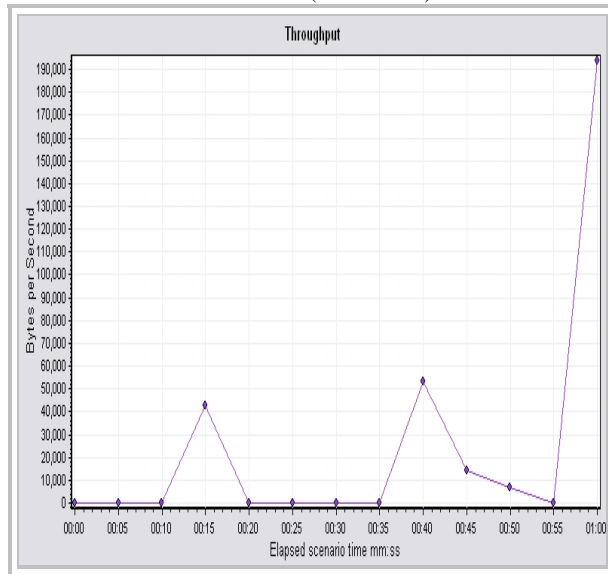
Measurement	Graph Min	Average	Graph Max.
Throughput	0.0	93721.631	387549

TABLE III
HITS PER SECOND (FIVE USERS)

Measurement	Min.	Ave.	Graph Max.	Graph Medium	Graph SD
Hits	0.0	4.712	13.4	4.8	4.282

Tables 1 to 5 and the associated figures display information such as the number of hits made on the web server as well as the throughput in terms of the number of bytes (sent and received). Such graphs help us evaluate the amount of load generated by users and the performance of the web servers.

FIGURE III
THROUGHPUT (SINGLE USER)



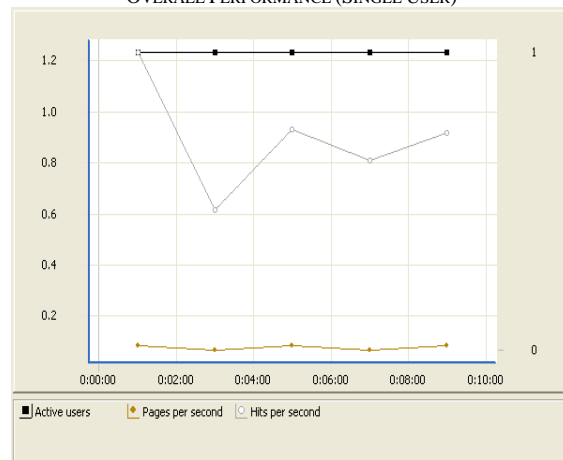
III. PERFORMANCE TESTING WITH WAPT (WEB APPLICATION PERFORMANCE TOOL)

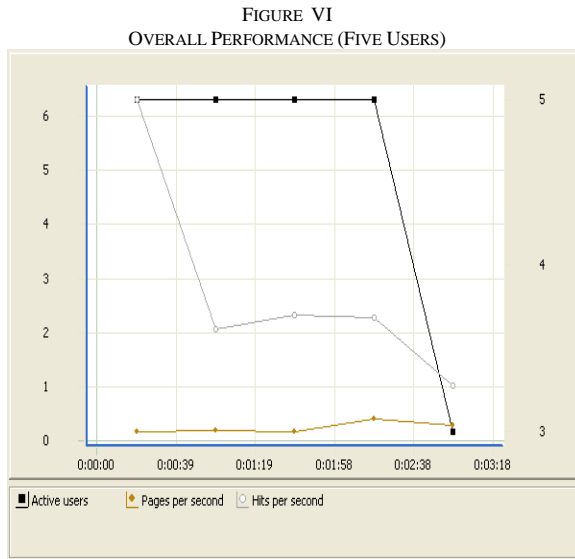
WAPT has been used widely to assess private business applications to a distributed web portals. The latter typically includes load balancers, web servers, application servers, database storages, etc. Applications are typically developed in Perl, ASP, HTML, PHP, Javascript and Java.

TABLE IV
THROUGHPUT (SINGLE USER)

Measurement	Min.	Ave.	Graph Max.	Graph Medium	Graph SD
Throughput	0.0	15720.98	193773.5	0.0	51923.19

FIGURE V
OVERALL PERFORMANCE (SINGLE USER)





IV. PERFORMANCE TESTING WITH OPENSTA

Open STA (Open System Testing Architecture) is an open source tool to test applications in a distributed. Resource utilization information from web servers, application servers, database servers etc under test can be monitored, graphed. Elapsed time and starter time provide valuable information and developers are encouraged to use this tool at the early stage of the development life cycle. For the same application, one can vary the number of virtual users to analyze the effect of loading. Alternatively, the same application can be developed in Perl, PHP, ASP, Java, etc. and compared. Such a study has already been reported in a previous IMECS conference in 2008. As mentioned in that paper, this tool is installed in the client machine and there is no need to disturb the servers.

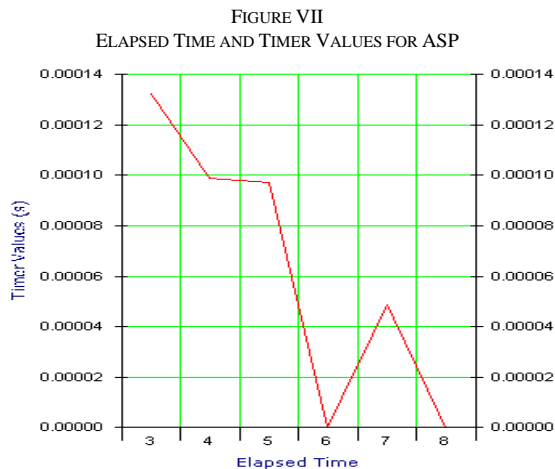


TABLE VI
STARTING TIMER VALUES FOR ALL APPLICATIONS

Serial Number	Application	Starting timer Value in sec
1	JSP	10
2	ASP	14
3	JavaScript	16
4	Perl	05
5	PHP	05

V. CONCLUSION

Load Tests are end-to-end performance tests under anticipated production load. The objective of such tests is to determine the response times for various time critical transactions and business processes. This will ensure compliance with documented expectations or Service Level Agreements. Load tests also measure the capability of an application to function correctly under load by establishing pass/fail/error rates; they supplement regression tests. Load Tests require substantial input from the business so that the anticipated business activity may be accurately simulated in a test environment. Proper user profiles are needed for the testing process, and this itself encourages proper understanding of the application. Load Tests should not only be conducted with today's production size database but also with projected future database expansion. The tools for such Load Tests should be reliable in the sense that if tests are repeated they give the same results. Today's websites are developed incrementally, and Load Tests should be included early in this iteration. Although tools such as openSTA have good graphical user interface, we do need to understand the nuances in setting parameters. The graph in section 4 required us to set scheduled task to 10 seconds and stop task to 5 seconds while using OpenSTA. Of course, the other important parameter is the number of virtual concurrent users.

REFERENCES

- [1] Aaron Don M. (2008) "Quantitative Evaluation of Open Source Content Management Systems", Africa IEEE Multidisciplinary Engineering Education Magazine, Vol. 3, No. 2.
- [2] COSMIC: "COSMIC-FFP Measurement manual", version 2.0.
- [3] Let us Learn & Master Load Runner 9.0.
- [4] Mercury Load Runner TM Tutorial Version.
- [5] Alan Boulanger, "Open Source Software Versus Proprietary Software".
- [6] Russ Mitchell, "Proprietary Software Can't Control the World".
- [7] R. Thirumalai Selvi, R.Meenakshi, N. V. Balasubramanian, and George T. Manohar, Performance Analysis of Proprietary and Non-Proprietary Software, Lecture Notes in Engineering and Computer Science: Proceedings of The International MultiConference of Engineers and Computer Scientists 2008, IMECS 2008, 19-21 March, 2008, Hong Kong, pp.982-984
- [8] Steven Weber, "Success of Open Source".
- [9] Louis Weitzman and Stephen Evenchick, "Using Open Source to Design and Develop".

APPENDIX – LOADRUNNER OBJECTS

Vuser Scripts	A Vuser script describes the actions that a Vuser performs during the scenario. Each Vuser executes a Vuser script during a scenario run. The Vuser scripts include functions that measure and record the performance of your application's components.
Load Test	Tests a system's ability to handle a heavy workload. A load test simulates multiple transactions or users interacting with the computer at the same time and provides reports on response times and system behavior.
Run-Time Settings	Run-Time settings allow you to customize the way a Vuser script is executed. You configure the run-time settings from the Controller or VuGen before running a scenario. You can view information about the Vuser groups and scripts that were run in each scenario, as well as the run-time settings for each script in a scenario, in the Scenario Run-Time Settings dialog box.
Scenario	A scenario defines the events that occur during each testing session. For example, a scenario defines and controls the number of users to emulate, the actions that they perform, and the machines on which they run their emulations.
Scheduler	The Schedule Builder allows you to set the time that the scenario will start running, the duration time of the scenario or of the Vuser groups within the scenario, and to gradually run and stop the Vusers within the scenario or within a Vuser group. It also allows you to set the load behavior of Vusers in a scenario.
Session	When you work with the Analysis utility, you work within a session. An Analysis session contains at least one set of scenario results (Irr file). The Analysis utility processes the scenario result information and generates graphs and reports. The Analysis stores the display information and layout settings for the active graphs in a file with an .lra extension. Each session has a session name, result file name, database name, directory path, and type.
Transactions	A transaction represents an action or a set of actions used to measure the performance of the server. You define transactions within your Vuser script by enclosing the appropriate sections of the script with start and end transaction statement.
Vusers	Vusers or virtual users are used by LoadRunner as a replacement for human users. When you run a scenario, Vusers emulate the actions of human users working with your application. A scenario can contain tens, hundreds, or even thousands of Vusers running concurrently on a single workstation.