

Performance Testing and Profiling of Web based Application in Real Time

Kajol Mittal

Assistant Professor
Department of CSE
ABESIT, Ghaziabad

Rizwan Khan

Associate Professor
Department of CSE
ABESIT, Ghaziabad

ABSTRACT

Performance testing and analysis process involves testing with the help of performance scripts and executing the scripts to run the application with the help of some performance test tool and monitoring the application performance using a APM Tool. Once the script is ready then these tests can run quickly and efficiently and Performance Analysis can be carried out by drilling down the data in APM Tool. Since the cost of performance testing is in the form of efforts and time required to create the scripts, not all tests can be converted to automated test. There should be a valid reason to pay that cost.

Performance Testing and Analysis plays an increasingly important role in the global economy and in daily experience. It helps maintain the application response time in order to prevent revenue loss. The objective of the paper is to create a pipeline by integration of Jmeter(Open Source Performance Test Tool), Dynatrace (Leading Application Performance Management Tool, Jenkins (Simple CI Server), This pipeline creation aims is to create continuous integration framework to automatically trigger performance test scripts and publish performance test results, dynatrace dashboard in Jenkins. Scripts are made using Jmeter and Application under Test is Easy Travel application.

Keywords

Performance testing, Performance analysis, continuous integration, application performance management

1. INTRODUCTION

“Testing is the process of executing a program with the intent of finding errors”- Glen Myers. Building a successful product depends on two fundamental ingredients — functionality and performance of a software product. 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, tasks and to furnish information rapidly and accurately despite high multi-user interaction or constrained hardware resources [2]. Performance Testing is done to determine the response time and throughput of any web application [5]. Performance testing is use to verify whether the system meets its non-functional requirements identified in SRS(Software Requirement Specification) document or not. 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.

Easy Travel provides a web portal which allows users to log

in, search for journeys to various destinations, and select promotional journeys directly that are offered and to book a journey using credit card details. Additionally a Business-to-Business (B2B) web portal for travel agencies is provided where travel agencies can manage the journeys that they offer and can review reports about made bookings. Easy Travel is a multi-tier application implemented in .Net and Java.

The need of the proposed system is to provide an online platform to book tickets easily via an online platform that can cater high web traffic during holiday season and provides a smooth user experience to its visitors, the platform should be able to handle high load without an increase in response times.

The paper aim is to automate the performance testing and performance analysis process in software development life cycle by integrating performance test tool (Jmeter) and performance analysis tool (Dynatrace) with CI Platform (Jenkins). Manual processes are slow, error-prone and can sabotage compliance efforts. This can eliminate them with automation. It makes the job simpler and easier. It reduces the time and number of resources.

Following are being covered in the papert:

- Automate Performance Test Execution on build deployment: To reduce manual effort. Without human interaction the Jmeter scripts will get automatically trigger on Jenkins machine
- Automate Performance Test Results Publish Process: To avoid manual intervention in Test Results publish process, The developers can directly log in to Jenkins and check performance test results generated from Jmeter performance test.
- Automate Performance Profiling: The Dynatrace results would be directly broadcasted to Jenkins and developers can have a quick look at the same, thereby reducing the time in gaining important insight to application performance at code and server level.

2. EXISTING AND PROPOSED SYSTEM

2.1 Present System

This paper aims to automate and integrate performance testing and performance analysis process by creating scripts which will be helpful in performing the performance test and integrating them with Jenkins which will automatically trigger the script when build is deployed, Also, Dynatrace Agents would be configured on the Easy Travel Application, The dynatrace cusum dashboards providing insight to application performance bottlenecks would be broadcasted to Jenkins CI Platform

The main objectives of the WLIQ Automation are:-

- To reduce the overhead of manually triggering Performance tests on each new build deployment.
- To aid in performance bottleneck identification using Dynatrace
- To help the developer access Performance Test and Performance Profiling data at a single platform

The scope of the paper is to overcome the below mentioned problems,

As per the Current approach, tester manually triggers performance test script from Jmeter and manually saves reports created in Dynatrace and Jmeter and shares the result via e-mail this takes 6-7 hours of time and efforts to complete it. On top of that if it is needed to re run the test it will again take few more hours to retrieve the required scripts and configure them to run. As tester and developer needs to repeat the same steps every time to perform the Performance testing and performance analysis, it is an overhead for the tester and developer to do this repetitive job.

2.2 Drawbacks of the existing system

- **Time consuming process:** The present system was mostly manual, it provided very less automation. Hence the entire process was time consuming.
- **Extensive need of manpower:** Since the present system was manual, it required extensive man power to handle the whole process. It requires user to download the build from the server, install the build and run the smoke test.

And also,

- Time consumption.
- Difficulty in data retrieval.
- Tedious manual work

To minimize these pitfalls, the continuous integration via Jenkins platform have been proposed.

2.3 Proposed System

The system proposed in this paper will automate the performance testing and performance analysis process in software development life cycle by integrating performance test tool (Jmeter) and performance analysis tool (Dynatrace) with CI Platform (Jenkins).

As soon as the new build is deployed, Performance Test Scripts will be triggered in Jenkins, Jmeter scripts are customized to broadcast the request to Dynatrace (Application Performance Management Solution), Dynatrace will capture these and collate with the available profiling data, Via a Jenkins plugin, Dynatrace data will be broadcasted to Jenkins CI Platform, where user can see the Jmeter test results as well as Dynatrace Dashboards.

2.4 Advantages of the Proposed System

The system has been proposed with the following objectives in mind. Some are:

- Significant savings in time.
- Ease of maintaining information.
- Accuracy of data.

The specific objectives that can be met by the proposed system are:

- Improved efficiency.
- Reduced processing time.
- Reliable

Key Benefits of Continuous Integration:

- Reducing risk
- Reducing overheads across the development & deployment process
- Enhancing the reputation of the company by providing Quality Assurance

3. SYSTEM ANALYSIS AND DESIGN

3.1 System Study

A wide range of methods was adapted to study the functionality of the product in testing perspective and the performance testing and analysis involved.

3.2 Observations

This existing system was observed for several period of time and it was found to have high response time

3.3 Discussions

As the observations are made a process of the proposed system discussion were held in the paper to develop the same.

3.4 Suggestions

Based on the requirements the information for developing the continuous integration is gathered from various users and the same is to be processed.

3.5 Documentation

The necessary details collected were documented for future references.

4. SOFTWARE REQUIREMENT SPECIFICATION

This paper has been designed using the Continuous integration Platform (Jenkins), Application Performance Profiling tools and Performance Test Tool and attempts to take a notable step towards Automation of the existing manual process. The continuous integration provides lot of process improvement and efforts reduction at different levels.

Taking into account, the problems associated with the present system, an attempt is made to increase the performance with the help of these automated process via CI to achieve accurate and reduction of overall process by 2 to 3 hours.

4.1 Requirements Specification

- **Purpose:** Continuous integration will operate in the environment to facilitate all the users in the proposed system can use the scripts on their period of test execution, and analyze data provided by Dynatrace to find possible performance bottlenecks in the system. The system has been implemented to provide accuracy, reduced manual intervention and time consumption.
- **Scope:** The “PERFORMANCE TESTING AND PROFILING OF WEB BASED APPLICATION WITH CONTINUOUS INTEGRATION (CI)” the system is used to provide the user to reduce the

manual execution of performance tests and reducing the time for finding performance bottlenecks whenever needed.

4.2 Design Constraints

Some of the factors that restrict the design of the system include Resource limits, Operational Environment and Policies. During the development process all these issues should be taken into consideration.

4.3 Process Model

The development for this proposed system follows the V-Process Model. The model is suitably customized according to the needs.

This depiction is requirements-driven, and starts with identification of user requirements. When these are understood and agreed-to, they are then placed under system control, and through decomposition the system concepts and system specification are developed. The decomposition and definition process is repeated over and over until, ultimately, lines of code and piece parts are identified. Agreement is reached at each level, and the decisions are placed under proposed system configuration management before proceeding to the next level. When the lowest level is defined, can move upward through the integration and verification process.

Principles and Objectives:

- Iteration among disciplines and
- Decision gates
- High level of detail at the beginning
- Early involvement of the team
- Continuous improvement
- Change Control regimen
- Analysis (on the right) Synthesis (on the left)
- To reduce risk
- Reduce Cycle Time
- To reduce uncertainty

Following are the different phases and the activities carried out in each phase for the proposed system:

- **Requirement Specification Phase:** This phase involves the study of the existing system and the analysis of the features needed, in the system to be developed. The functional & non-functional requirements of the system to be developed were identified and documented in the Software Requirement Specification (SRS) completely and unambiguously.
- **Functional Specification Phase:** Based on the

SRS document the functional specification document was prepared. Functional specification document details all the functionalities and the functional requirements of the system to be developed. Also system test cases were prepared based on the FS document. System Test Cases explain the approach for testing the features and functionality of the proposed system during System Testing.

- **Design Phase:** Based on the Functional specification document, the design document was prepared. The Design document details the process flow of each of the modules to be designed. The screen designs of all the modules have been captured in the design document.
- **Coding Phase:** Based on the design document, the source code for all the modules was written. The code was compiled and errors were fixed. Unit test cases were prepared explaining the approach for testing the implementation of features and functionality of each component or each module of the system.
- **Unit Test Phase:** The source code was tested using the Unit Test Cases. The results of the test cases were compared with the expected results and the defects found were resolved by making suitable modifications to the source code. Unit Test Cases were updated where necessary.
- **System Test Phase:** Integration Testing was combined with System Testing, as integration of modules can be handled in System testing. The system was tested as a whole using the System Test Cases. The results of System Testing were recorded. All the defects found during the system testing were resolved by making suitable modifications to the source code. The System Test Cases were updated where necessary.
- **Performance Testing Phase** The Performance Test Scripts shall be triggered from Jenkins CI Platform utilizing Jmeter and jenkins integration and application performance will be monitored using jenkins utilizing Dynatrace

5. DESIGN SPECIFICATION

This system has been designed using the scripting and attempts to take a notable step towards Automation of the existing manual process. The automated script provides lot of accuracy at different levels to users. In addition, it is designed to provide dynamic logs, thus eliminating the user to manually log the executions.

Taking into account, the problems associated with the present processing system, an attempt is made to decrease the manual effort by increasing the logging process with the help of these automated scripts to achieve accurate logging and reduction.

5.1 Development Environment

The Scripts are developed in jmeter and Performance profiling is via dynatrace. For continuous integration Jenkins is used.

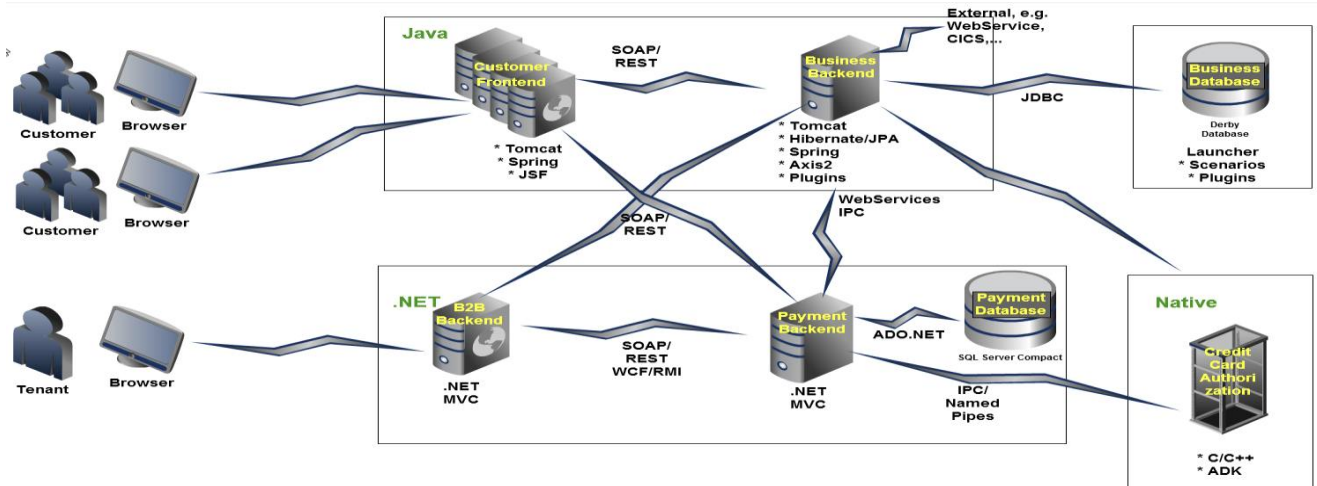


Fig 1: System Overview

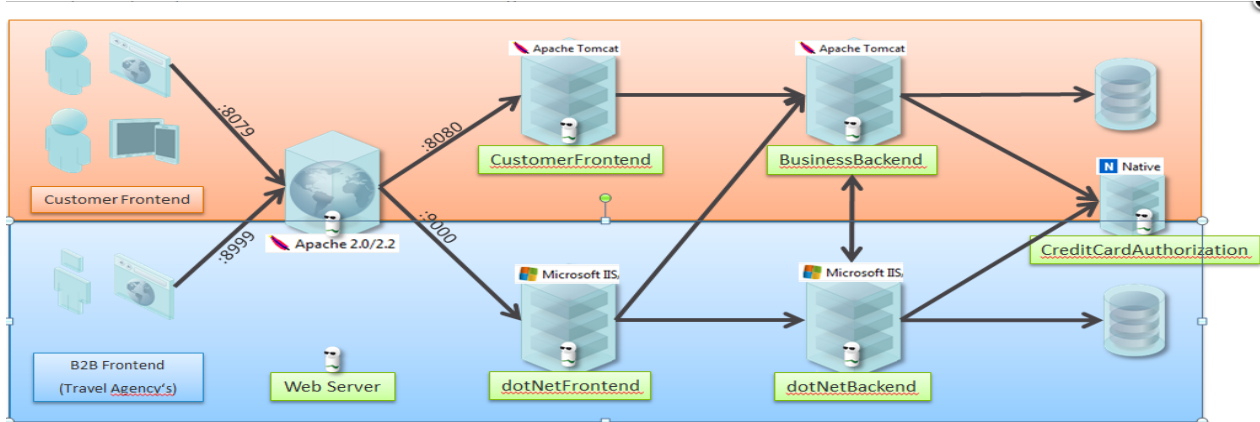


Fig 1: System Description

5.2 Business Requirement

According to the current system requirement this PERFORMANCE TESTING AND PROFILING OF WEB BASED APPLICATION WITH CONTINUOUS INTEGRATION (CI) will take care of executing the performance test scripts whenever it is invoked, it reduces the amount of time and intervention made by the user in order to execute the manual steps and proper logging of all the steps is carried out throughout the execution, Dynatrace results are broadcasted to Jenkins and Dashboards are easily accessible to developers for their perusal, Manual effort is reduced and able to reduce the overall time taken for performing the performance bottleneck analysis. Utilizing the process time effectively, and the time and resources that spent on manual can be utilized for other activity.

5.3 System and Functional Requirements

The system will be designed in a modular fashion to provide easy integration and also make it easier to incorporate changes later on in the system.

5.4 Non-Functional Requirements

5.4.1 Performance

This platform should not show high response times in critical business transactions.

5.4.2 Customizability

The system has to be a customizable one. This should be customizable to allow the user to run the test script based on

the requirement. Add/remove any script as per the user requirement.

5.4.3 Reliability

The system should not go down during working hours.

6. TESTING METHODOLOGIES

The scope of testing includes testing the performance of the critical business functionalities against the requirements as specified in the requirement analysis and design document. The test data required for the individual subsystems will be prepared beforehand and made available for performance testing the systems.

Functional Requirements				
I D	Associate d ID	Nonfunctiona l Requirement	Activities	Scenario covered
1	1.1	Login functionality response time should be less than 5 sec for 50 users Load Test		Login
2	2.1	Browse functionality		Browse

		response time should be less than 5 sec for 50 users Load Test		
3	3.1	Book Ticket functionality response time should be less than 5 sec for 50 users Load Test		Book Ticket

Non Functional Requirements				
ID	Associate ID	Functional Requirement	Activities	Scenario covered
1	1.1	User Must be able to see application profiling data for Dynatrace Agents configured in Jenkins		Dynatrace integration with Jenkins
2	2.2	The Jmeter script should trigger when “Build Now” button is pressed in Jenkins		Jmeter integration with Jenkins
3	3.3	The Jmeter results file should be available in Jenkins		Jmeter integration with Jenkins
4	4.4	The Dynatrace Dashboard data should be visible in Jenkins		Dynatrace integration with Jenkins

7. CONCLUSION

The objective of the paper is to create a pipeline by integration of Jmeter(Open Source Performance Test Tool), Dynatrace (Leading Application Performance Management Tool, Jenkins (Simple CI Server), This pipeline creation aims is to create continuous integration framework to automatically trigger performance test scripts and publish performance test results, dynatrace dashboard in Jenkins

8. REFERENCES

- [1] Ms. S. Sharmila¹, Dr. E. Ramadevi², “Analysis of Performance Testing on Web Applications” in International Journal of Advanced Research in Computer and Communication Engineering Vol. 3, Issue 3, March 2014.
- [2] Shikha Dhiman, Pratibha Sharma “Performance Testing: A Comparative Study and Analysis of Web Service Testing Tools” in IJCSMC, Vol. 5, Issue. 6, June 2016, pg.507 – 512
- [3] Rina and Sanjay Tyagi, “Comparative study of performance testing tools”, International Journal of Advanced Research in Computer Science and Software Engineering , 2013.
- [4] 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.
- [5] Shagun Bhardwaj, “Performance Testing Tools: A Comparative Analysis”, International Journal of Engineering Technology, Management and Applied Sciences, Volume 3 Issue 4, April 2015.
- [6] Harsh Verma, Arun Krishnan and Sivasubramanian. Analytics Performance Load Test. International Journal of Computer Applications 176(2):10-13, October 2017.
- [7] Weyuker, E.J., Vokolos, F.I., 2000. Experience with performance testing of software systems: issues, an approach, and case study. IEEE Trans. Software. Eng. 12, 1147—1156
- [8] Zhu, K., Fu, J., Li, Y., 2010. Research the performance testing and performance improvement strategy in the web application. In: 20102nd International Conference on Education Technology and Computer, vol. 2.
- [9]