ISSN No:-2456-2165

Automated Web Application Loaded Testing System with AppPerfect Load Test

Khin Than Nyunt
University of Computer Studies (Mandalay), Myanmar

Abstract:- Nowadays, a web application with high response time and poor performance can lead to huge problems. So, I want to propose Web Load Testing System to avoid these problems. Web Load Testing will help in identifying the performance problems before we deploy our web applications for end users. And then, design and simulate usage traffic which can be used to test our web application infrastructure for performance (response times), reliability (availability and integrity) and scalability (throughput). Load Testing of web applications involves testing the performance and scalability of our web application with varying user load. Moreover, involves simulating real-life user load for the target web application. Concurrently, determine how our web application behaves when multiple users hits it simultaneously. There are several load testing tools for web application available in the open source nowadays that are free of charge. One of them is AppPerfect Load Test Tool is used in this systemthat provide a script recorder that allow to easily record user actions into a script through graphical user interface and replay the scripts simultaneously as a load test and analyze the parameterization and validation process as user define.Whenever, scenariossuch as large number of concurrent users, large number of URLs, sustained usage over hours/days, etc. can be tested using HTTP/HTTPS and JDBC protocols.

Keywords:- *Automated Testing; Web Application Load Testing Approach; AppPerfect Load Test Tool.*

I. INTRODUCTION

Nowadays, every organization and people who interest of web application is using the internet to conduct their online business or society or other studies. There is necessary to rapidly get their requested / received webpages to finish for their work completely done soon without a page takes a long time to load in the memory. Because more and more the usage of internet with web application is more and more increase in the age of Information Technology that we can see own websites and pages such as political environment, business environment, education environment and social environment anywhere, we need to know more and more knowledge about the performance and reliability and scalability of our web application with varying user load. Therefore, it is important to understand the performance of web application for any organization and people. So, necessary to know how the load testing is used to test performance of a website. Introduce to this system to distribute automated load testing of web application using AppPerfect load test tool. Generally, the process of load testing is to create the test script, to record a test, to execute the recorded test, and by parameterization and validation the group / tasks. Moreover, to execute the scenarios and authentication mechanism and to repeat playback the recorded test from multiple numbers of virtual users.

In this system, there are many aspects of application's quality of services in automated testing that can run unattended heavy usage and also provides high test coverage. And then, that is highly scalable load virtual user's simulation process thatcan be distributed over multiple machines. Moreover, cost effective, reliable, efficient and highly extensible and also makes any platform can be used to playback over different hardware configuration for testing.

II. WEB APPLICATION LOAD TESTING APPROACH

This system intend to know what is load testing, why is load testing important, and how to get started with load testing using AppPerfect load testing tool and how the system behaves with a large number of users and what is the response times, throughput rates and resources-utilization levels received/sent for pages under different scenarios. Throughput means thatcapable of to handle multiple transactions in a given period time and represents the number of request transaction in specified time duration. As the number of concurrent users increase, the throughput increase almost linearly with the number of request. Response times that it also important to find out how much time each of transaction process to complete and it increases proportionally to the user load. A response time is also defined as the delay between the point of request and the first response.Actually, load testing is a kind of performance testing which determines a system's performance under real-life load conditions and measures the website's quality of services performance based on actual customer behaviors.When the user visit web site, a script recorder records the communication and then creates related interaction scripts then executes the recorded test. A load generator tries to replay the recorded scripts, which could possibly be modified with different test parameters and capable of validation process before replay. In the replay procedure, both the hardware and software statistics will be monitored and collected, these statistics include the CPU, memory, disk IO of the physical servers and the response time, throughput of the system under test. And at last, all these statistics will be analyzed and a load testing report will be generated.see “Figure. 1”

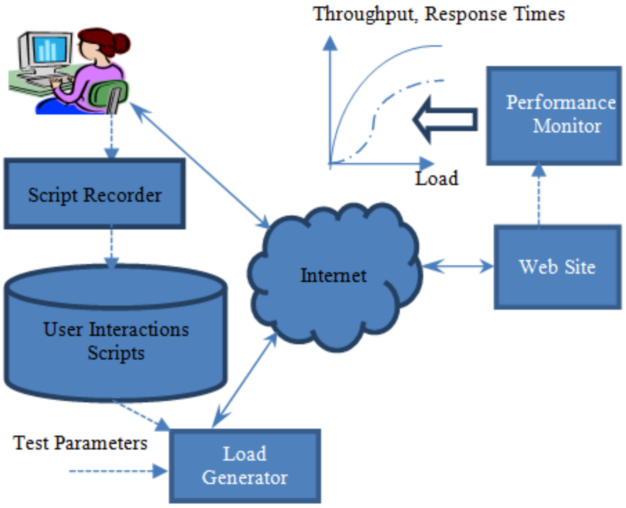


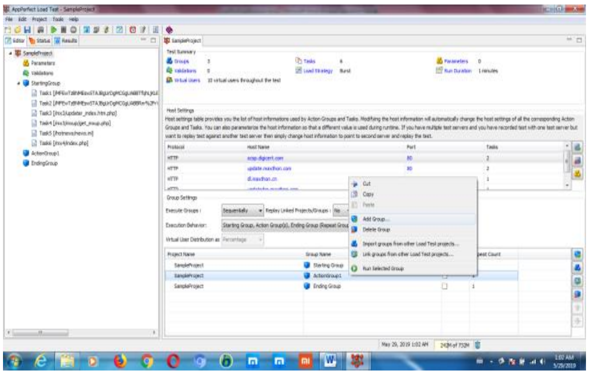
Fig 1:- Web Load Testing Process

III. THE PROCESSING OF THE SYSTEM

Firstly, we will create three main groups such as Starting group, Action group and Ending groupfor Web Load Testing.There are two strategies in starting group which are “group details” and “login user setting”. The group details consist of group name, host name, path name in which we will enter the correct name respectively and define repeat count and virtual users. Moreover, three checkbox which are “pause execution of group”, “stop execution of group”and “ignore group”. We can click these checkbox as user desired “N” iterations times in drop down list box. Login user setting involves group contains login task checkbox which can check the task to enter login and authentication setting in which we will choose whether or not authentication by parameterizing.

The next one is Action Group in which we will also specify and choose group details, login user setting and authentication setting respectively as Starting Group. We can set login name and password to check the authenticity of the user to restrict the web page. Then, we can select the required authentication mechanism such as “No Authentication”, “Basic Authentication”, “Digest Authentication”, and “NTLM Authentication” from the authentication combo box in recording dialog box that provide a valid Username and Password and Domain name and then record the test.We can create several Action Groups in a single load test project and create a different set of tasks for each of these Action Groups. We can also add more Action Groups to our project edit and delete existing ones and execute complete Action Group even if duration of test has completed and also close HTTP connections after each iteration and create new connection for successive iteration. We can also Cut, Copy, and Paste Action Groups.Moreover, Action Group is used by all types of Load Test tests: Web test/ Database test.

The last one is Ending Group which is logout task it may contain the tasks that will clear up database and perform logout operations etc. The Ending Group will be executed all the tasks for two types such as time-based test, and hits-based test. It is possibility that for run duration specified in terms of time for time-based test.The minimum value necessary for a time-based test is 1 minute and the maximum we can test for is 9,999 days. We would use a large number of hits and then how much time the specified virtual users will take to hit the specified number of time for hits-based test. The minimum value necessary for a hits-based test is 1 hit and the maximum we can test for is 999,999,999 hits.We can specify the length of the test in terms of either time duration or numbers of hits. We can also specify many 100 virtual users and 20 few hits throughout the test see “Figure. 2.”


AppPerfect Load Test - SampleProject
File Edit Project Tools Help
Editor Status Results
SampleProject
Parameters
Validations
StartingGroup
Task1
Task2
Task3 [/mx3/updater_index.htm.php]
Task4 [/mx3/moup/get_moup.php]
Task5 [/hotnews/news.m]
Task6 [/mx4/index.php]
ActionGroup1
EndingGroup
Cut
Copy
Paste
Add Group...
Delete Group
Import groups from other Load Test projects...
Link groups from other Load Test projects...
Run Selected Group

Test Summary

Groups 3 Tasks 6 Parameters 0
Validations 0 Load Strategy Burst Run Duration 1 minutes
Virtual Users 10 virtual users throughout the test

Host Settings

Host settings table provides you the list of host informations used by Action Groups and Tasks. Modifying the host information will automatically change the host settings of all the corresponding Action Groups and Tasks. You can also parameterize the host information so that a different value is used during runtime. If you have multiple test servers and you have recorded test with one test server but want to replay test against another test server then simply change host information to point to second server and replay the test.

Protocol	Host Name	Port	Tasks
HTTP	ocsp.digicert.com	80	2
HTTP	update.maxthon.com	80	2
HTTP	dl.maxthon.cn		1

Group Settings

Execute Groups : Sequentially Replay Linked Projects/Groups : No
Execution Behavior: Starting Group, Action Group(s), Ending Group (Repeat Group
Virtual User Distribution as Percentage

Project Name	Group Name		Repeat Count
SampleProject	Starting Group		
SampleProject	ActionGroup1	□	1
SampleProject	Ending Group	□	1

May 29, 2019 1:02 AM 242M of 732M

Fig 2:- Creating Starting Group, Action Group and Ending Group for Load Test

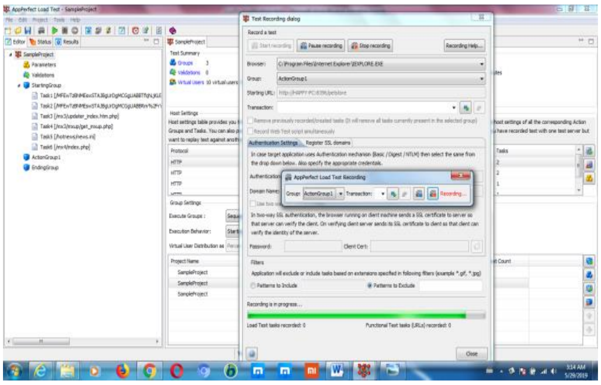
This system supported a number of execution load strategies to execute our test scripts. Each load strategy and its behavior will appear in the dialog box. The first two Strategies (Burst and Loop) work constant number of virtual user and the remaining strategies are all a variation of the Incremental strategy where the user count increases.We can choose desired six load strategies for the following:

- Burst : to test application for sudden loads, performs at random interval of times
- Loop: to test application for sequential load a fixed number of times, runs till the loop count
- Incremental : to test application to reach its breaking point when the load increases gradually, incremented by user value, time value and maximum number of users
- Exponential: to test application to reach its breaking points but here, the load increases in leaps, don't have to provide an increment by value as the user load doubles every increment by time period.
- Quadratic : to test application’s recovery well from a period of increasing loads, peak traffic
- Cyclic means: to test application’s with cyclic loadon its during the test

Host settings table provides the list of host information used by Action Groups and Tasks. Modifying the host information will automatically change the host settings of all the corresponding Action

Groups and Tasks. We can also parameterize the host information so that a different value is used during runtime. If we have multiple test servers and recorded test with one test server but want to replay test against another test server then simply change host information to point to second server and replay the test. In the Group setting, we must choose sequentially or simultaneously execute the group and whether replay or not the linked project/group and execution behavior and virtual user distribution as percentage.

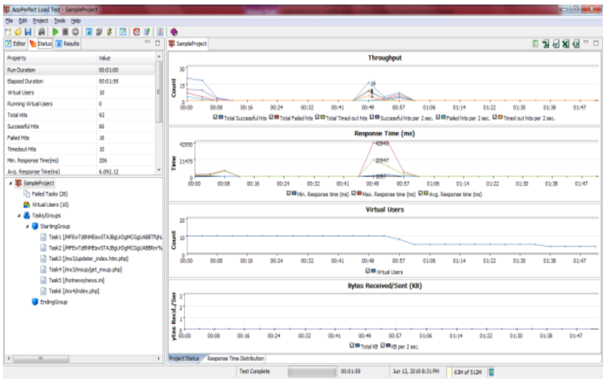
Secondly, after a project is defined or created, the next step is to record a test. Recording a test essentially means defining a set of URLs that will fire at our target application. To start recording a test we will click recording icon when the time “Record a test” dialog box will appear and lunch.We must be select “ActionGroup1” in the group combo and regarding the record there are three main button “Start recording” to record a test, “Pause recording” to pause a recorded test and “Stop recording” to stop a recorded test. A test can be recorded very simply by using a Web Browser. In this system, an instance of Internet Explorer will be lunched and the web page corresponding to the starting URL will be loaded in the browser. Then, navigate through the web application to record few URLs that the URLs navigated will get added to the ActionGroup1 node as tasks. Once the recording is complete close the IE browser and click on Stop Recording button in “Record a test” dialog box to stop recording and click Close button to close the “Record a test” dialog box. After recording the test we will execute the test. To execute the test,when the time we can click “Run” icon to start load testing here the current status of the test execution will be displayed. Various test metrics such as throughput, response time, virtual user count and bytes received/sent etc., are continuously refreshed in the charts. The test will run for the duration specified in Load Test project properties dialog box. Once the test is completed, the display will switch to the “Result” tab and test results will appear. We can view various result summaries by clicking on the items in the tree on the left. We can see “Figure. 3.”



Browser: C:\Program Files\Internet Explorer\IEXPLORE.EXE
Group: ActionGroup1

Fig 3:- Recording Test

After recording the test, we will see the result with Throughput, Response Time (ms), Virtual Users and Bytes Received/Sent (kb) in “Result” view see “Figure. 4.” In addition to regarding the response time, we can specify that if during execution response time of the task is outside of the range specified then should application ignore the hit or should replace the response time for that task with the range specified. To affect the response time from our application we must be select the browser whose behavior we wish to simulate and can set the percentage of users that will simulate for Internet Explore and Mozilla Firefox. Also need to decide whether we want to clear the browser cache after each iteration. And also need to select the bandwidth settings we wish to simulate. We must notice, “throtting” is not applied to the test by default. If we want to limit the requests fired according to the bandwidth, then we can select the checkbox “Throttle test based on Network Bandwidth”. And then, if we want to test the load balancing feature of our application server, we can do so by providing a list of valid IP addresses.



Property	Value
Run Duration	00:01:00
Elapsed Duration	00:01:59
Virtual Users	10
Running Virtual Users	0
Total Hits	92
Successful Hits	66
Failed Hits	16
Timedout Hits	10
Min. Response Time(ms)	206
Avg. Response Time(ms)	6,092.12

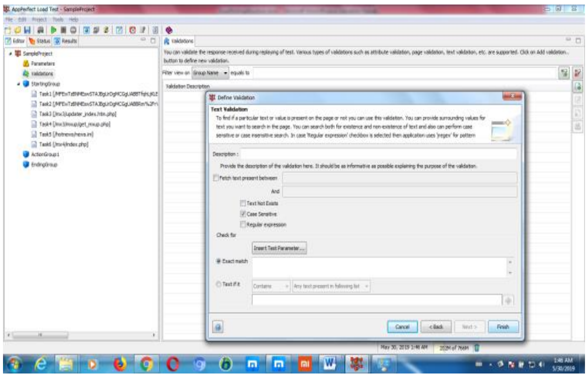
Fig 4:- Load Test Result

Furthermore, we can get Group Summary, Response Code and Response Time respectively with html, pdf, csv, xls and xml format see “Figure. 9, 10 and 11.”While the system calculates success or failure of a request based on basic response information such as Response Code and Response Time, but sometimes its not sufficient. If the task is successful, the user request is complete and save Successful Response details in this system.If the task is fail, will show the error that is connection timeout or remote server down or need an external proxy to be configured. At the time we will stop execution of test if number of failed task count exceeds thatspecify the maximum number of tasks that can fail before the entire test execution is stopped. So,we will specify the maximum count either for an individual virtual user or for the entire test as an aggregate value. We can see the Top(N) tasks taking maximum response time provided the default value is 10 tasks and change this to view more tasks that take high and maximum times to provide the responses and the next Fail (N) tasks provided the default values is 25 tasks and virtual users at the left top corner of project respectively.

Thirdly, validation can only be defined after a test has been recorded that will automatically check a URLs Http response code to ensure there was no failure, often times it is necessary to check the html response for certain values to validate the application worked correctly. Its often required to check whether the replayed pages are indeed the one that was recordedin the

ISSN No:-2456-2165

recording process or whether there is any change in the web page content that has happened after the web pages were recorded to achieve by settings validations for the page. To set validation click on the "validation" node in the Editor tree and select the required validation option that provide various types of validations such as attribute validation, page validation, text validation, date validation, database query validation are supported to define new validation. And also we can filter any validation by choosing "group name" or "Task name". We can see text validation see"Figure. 5."



You can validate the response received during replaying of test. Various types of validations such as attribute validation, page validation, text validation, etc. are supported. Click on Add validation button to define new validation.

Define Validation

Text Validation

To find if a particular text or value is present on the page or not you can use this validation. You can provide surrounding values for text you want to search in the page. You can search both for existence and non existence of text and also can perform case sensitive or case insensitive search. In case 'Regular expression' checkbox is selected then application uses 'regex' for pattern

Description :

Provide the description of the validation here. It should be as informative as possible explaining the purpose of the validation.

□ Fetch text present between

And

□ Text Not Exists

☑ Case Sensitive

□ Regular expression

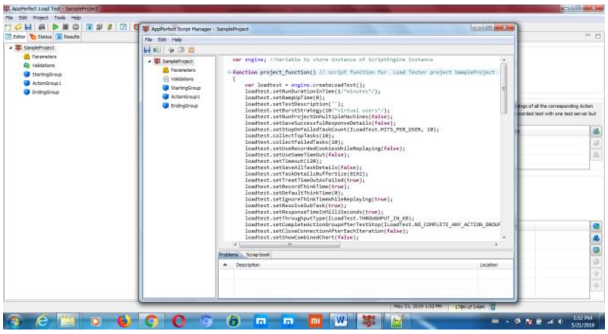
Check for

● Exact match

○ Text if it Contains Any text present in following list

Fig 5:- DefiningText Validation for Test

For custom validation on web page, users need to store recorded test as JavaScript and can add or edit custom code to the test script to perform any custom validation in "Script Editor" tab. Then, we can parse and fetch the appropriate field from the "define validation" dialog box and validate the required value. Regarding this, we can also use scripting APIs to fire any database query and perform the required validation by fetching the required data using the result set see "Figure. 6."

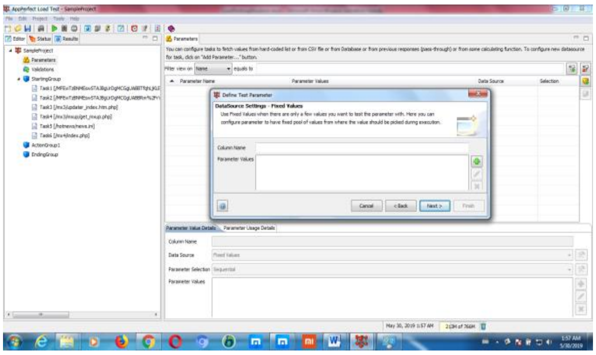


```
var engine; //Variable to store instance of ScriptEngine Instance

function project_function() // script function for  Load Tester project SampleProject
{
    var loadtest = engine.createLoadTest();
    loadtest.setRunDurationInTime(1/*minutes*/);
    loadtest.setRampUpTime(0);
    loadtest.setTestDescription('');
    loadtest.setBurstStrategy(10/*virtual users*/);
    loadtest.setRunProjectOnMultipleMachines(false);
    loadtest.setSaveSuccessfulResponseDetails(false);
    loadtest.setStopOnFailedTaskCount(ILoadTest.HITS_PER_USER, 10);
    loadtest.collectTopTasks(10);
    loadtest.collectFailedTasks(10);
    loadtest.setUseRecordedCookiesWhileReplaying(false);
    loadtest.setUseSameTimeOut(false);
    loadtest.setTimeout(120);
    loadtest.setSaveAllTaskDetails(false);
    loadtest.setTaskDetailsBufferSize(8192);
    loadtest.setTreatTimeOutAsFailed(true);
    loadtest.setRecordThinkTime(true);
    loadtest.setDefaultThinkTime(0);
    loadtest.setIgnoreThinkTimeWhileReplaying(true);
    loadtest.setResolveSubTask(true);
    loadtest.setResponseTimeInMilliSeconds(true);
    loadtest.setThroughputType(ILoadTest.THROUGHPUT_IN_KB);
    loadtest.setCompleteActionGroupAfterTestStop(ILoadTest.NO_COMPLETE_ANY_ACTION_GROUP
    loadtest.setCloseConnectionAfterEachIteration(false);
    loadtest.setShowCombinedChart(false);
```

Fig 6:- Custom Validation Script

Fourthly, as the test is recorded, all parameters used by this application are recorded automatically. Thus, in most cases, we don't have to manually create each parameter. After recording the test, simply click on the "Parameters" node in the Editor tree to view the recorded parameters. Moreover, many parameters are supported such as runtimes parameters, pass-through parameters, fixed parameters and parameterization in this system. Among them, parameterization depending on the number of iterations, multiple parameter values can be automatically tested with options to select parameters sequentially or randomly. Parameter values can be passed as Fixed value, or it can be Imported from a database, or Imported from a CSV file, or the user can define a range from which the values are to be picked by using the Calculated Values options. Parameter values can be passed in a sequentially or randomly. To configure new data source for task, click on "Add Parameter" buttonsee "Figure. 7."



You can configure tasks to fetch values from hard-coded list or from CSV file or from Database or from previous responses (pass-through) or from some calculating function. To configure new datasource for task, click on "Add Parameter..." button.

Define Test Parameter

DataSource Settings - Fixed Values

Use Fixed Values when there are only a few values you want to test the parameter with. Here you can configure parameter to have fixed pool of values from where the value should be picked during execution.

Data Source: Fixed Values

Parameter Selection: Sequential

Fig 7:- Defining Test Parameters

Anymore, we can also run the project / tasks simultaneously on multiple machines not to run sequentially. This system provides a unique view of the target machine with integrated Load Test, Stress Test and Performance Test parameters such as response time, throughputs and hit count along with monitoring of system resources such as CPU, Logical Disk, Physical Disk, Network Interface, Physical Memory and Virtual memory.We can see "Figure. 8."

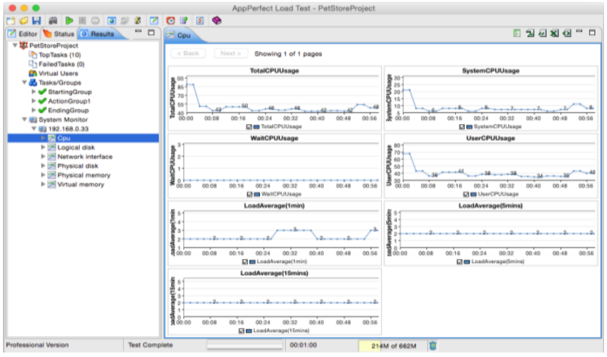
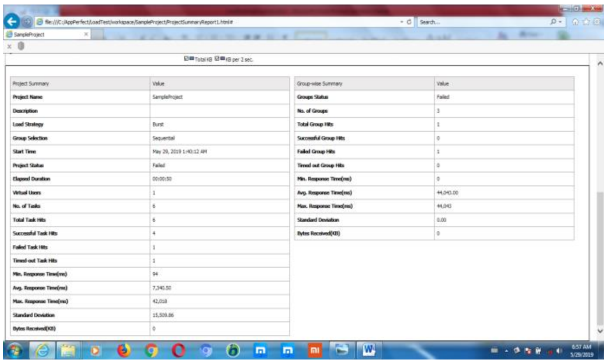


Fig 8:- Monitoring the System Resources

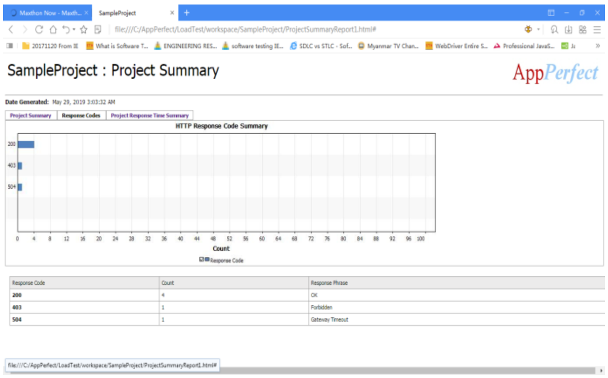
At the end of test execution resultswe can also view the detailedload test results such as Project Name, Description, Load Strategy, Group Selection, Start Time, Project Status, Elapsed Duration, Virtual Users, Number of Tasks, Total Task Hits, Successful Task Hits, Fail Task Hits, Timed-out Task Hits, Minimum Response Time (ms), Avg. Response Time (ms), and Maximum Response Time (ms) etc. for each of the virtual userexport them to HTML formats in Project Summary Tabsee "Figure. 9." And also we can see HTTP Response Codes Summary with Response Code, Count and Response Phrase in Response Code tab respectively see "Figure. 10." Andalso meet Response Time Summary with Response Time Interval (ms) and Frequency in Project Response Time Summary tab relatively see "Figure.11."


file:///C:/AppPerfect/LoadTest/workspace/SampleProject/ProjectSummaryReport1.html#
SampleProject
Search...
Total KB
KB per 2 sec.
6:57 AM
5/29/2019

Project Summary	Value
Project Name	SampleProject
Description	
Load Strategy	Burst
Group Selection	Sequential
Start Time	May 29, 2019 1:40:12 AM
Project Status	Failed
Elapsed Duration	00:00:50
Virtual Users	1
No. of Tasks	6
Total Task Hits	6
Successful Task Hits	4
Failed Task Hits	1
Timed-out Task Hits	1
Min. Response Time(ms)	94
Avg. Response Time(ms)	7,340.50
Max. Response Time(ms)	42,018
Standard Deviation	15,509.86
Bytes Received(KB)	0

Group-wise Summary	Value
Groups Status	Failed
No. of Groups	1
Total Group Hits	1
Successful Group Hits	0
Failed Group Hits	1
Timed out Group Hits	0
Min. Response Time(ms)	0
Avg. Response Time(ms)	44,043.00
Max. Response Time(ms)	44,043
Standard Deviation	0.00
Bytes Received(KB)	0

Fig 9:- Monitoring Project Summary

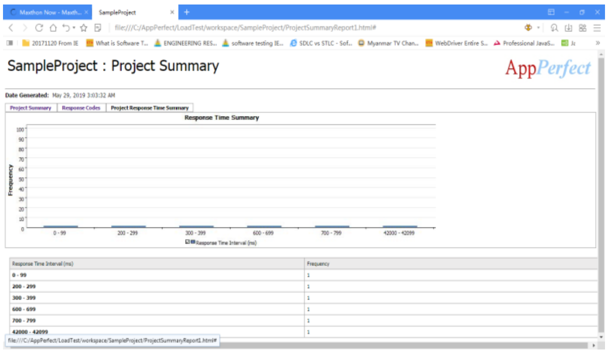


SampleProject : Project Summary

Date Generated: May 29, 2019 3:03:32 AM

Response Code	Count	Response Phrase
200	4	OK
403	1	Forbidden
504	1	Gateway Timeout

Fig 10:- Monitoring Response Codes



SampleProject : Project Summary

Date Generated: May 29, 2019 3:03:32 AM

Response Time Interval (ms)	Frequency
0 - 99	1
200 - 299	1
300 - 399	1
600 - 699	1
700 - 799	1
42000 - 42099	1

Fig 11:- Monitoring Project Response Time Summary

IV. CONCLUSIONS

This system is a fully automated Load Test, Stress Test and Performance Test Solution that is easy to use and cost effective.This system can design and simulate the number of concurrent user's requests to application can receive and meet the current infrastructure sufficient for user's demand. And then, this application scale access thousands of users and then this web application's response times and throughput effectively change as number of users increase use. All of these facts, this system is assumed to be most affordable, cost effective, efficient, reliable and accurate, increase customer satisfaction, reduce risk of downtime, improve scalability, improve deployment quality and maximize marketing campaign fund with Web Load Testing solution by users.

ACKNOWLEDGMENTS

I would like to special thank Rector, U KyawZwarSoe and Pro-Rector, Dr. San San Tint of University of Computer Studies (Mandalay) and also especially like to thank Head of Department of Information Technology Supporting and Maintenance (ITSM), Dr. KhinPhyo Thant and Department of Administration Affair and everyone.Finally, I am grateful to my Parents and Husband, U Tint Lwin, thank you for your love, support and energy, and appreciation forthis.

REFERENCES

[1]. Rijwan Khan, MohdAmjad, "Performance testing (Load) of Web Applications based on test case management" New Dalhi, India, 2016
[2]. Milad Hanna,Nahla El-Haggar ,Mostafa Sami "A Review of Scripting Techniques Used in Automated Software Testing," International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 5, No. 1, 2014
[3]. Bc. DávidChmurˇciak "Automation of regression testing of web applications" Brno, spring 2013
[4]. Dominique Guinard "A Web of Things Application Architecture -Integrating the Real-World into the Web" Switzerland, 2011
[5]. Daniel A. Menasce`, "Load Testing of web Sites" RAAS, 2013
[6]. Zhang Hu-li, ZhaungShu, "Research of Load Testing and Result Application Based on LoadRunner", National Conference on Information Technology and Computer Science (CITCS 2012), China
[7]. Scott Barber, Colin Mason,"Web Load Testing for DUMIES" Compuware Special Edition
[8]. Akshay Kumar Magapu Nikhil Yarlagadda "Performance, Scalability, and Reliability (PSR) challenges, metrics and tools for web testing" Sweden, 2016
[9]. Gustav Murawski, Philipp Keck, Sven Schnaible , "Evaluation of Load Testing Tools", 2014
[10]. MarijnHaverbeke, "Eloquent JavaScript" 3rd edition
[11]. Nicholas C. Zakas, "Wiley Professional JavaScript for Web Developers", 3rd Edition
[12]. BhartiBhattad, Dr. Abhay Kothari , "Study of Defects, Test Case andTesting Challenges in Website Project UsingTechniques", 2014
[13]. Zhen Ming Jiang, "Automated Analysis of Load Testing Result" January, 2013
[14]. http://www.appperfect.com
[15]. https://joomla.abstract.bg/automated_load test_stress_test_performance_test.pdf