ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 11, November 2018

Freeware Framework for Performance Testing of Software Web Applications

Mohammed Abdul Razack Maniyar[1], Suresh Koppula[2], Dr. Mohd Khalid Mubashir Uz Zafar[3]

Ph.D Research Scholar, Department of Physics, Dravidian University, Kuppam, India[1]

Senior Test Engineer, Radiant Sage India Tech Services Pvt Ltd., Hyderabad, India[2]

Associate Professor, Maulana Azad National Urdu University, Hyderabad, India[3]

ABSTRACT: Performance of software web applications has become an important factor in the software development life cycle. The present study gives a cost-effective freeware performance testing framework which caters the end-to-end performance testing requirements. The different phases of the performance testing life cycle of the software web applications are automated using open source tools. These tools are user friendly and can be easily integrated within the defined framework. The identified tools are freely available and compatible with the web-based technological applications. As this automated framework uses open source tools, it is a very cost-effective one as compared to the automation framework with vendor based licensed tools. This framework helps in meeting all the performance testing objectives and supports small, complicated and large scale enterprise systems.

KEYWORDS: Performance Testing Framework, Open source tools, Performance bottlenecks, Web application testing, Performance testing tools, Server Monitoring, Performance testing analysis.

I. INTRODUCTION

The current trend of using next generation platforms with new technologies and complex architectures has increased the risk of software application performance and also any application with high volume usage has to undergo Performance testing. Performance testing exposes the load related defects and various latent issues like Memory leaks, Threading problems and other longevity issues along with the most common response times and throughput issues. It is also seen that disruptive services with a lot of hic-ups impact the Business, customer loyalty, brand image and also the revenue. Sensing the importance of these, most of the software development organizations have started including Performance testing in the software development life cycle of the applications or products. This would help in optimization of various products, platforms, devices and technologies. It also gives enhanced experience and quality from a user's perspective and assures customer satisfaction. Performance testing also adds up Business value with better Business outputs with seamless performance.

A Performance Testing Framework with automated tools can ease the job and fasten the performance testing life cycle process. An End-to-End Performance Testing Framework in place would help in standardization and acceleration of testing process. Many other frameworks are available which uses vendor or licensed tools, which is expensive. This paper talks about a freeware framework which can be used for Performance testing of software web applications. This is much easier and simple and cost effective. This Framework with Freeware's enables repeatedly building new load tests without recording a new script thereby eliminating the additional work that may have been caused by changes to the UI or application. This Framework can be used for different types of Performance tests like Load, Stress, Spike and Endurance testing.

The framework presented here consists of Open Source tools which are easily available and can be easily implemented and integrated to cater the performance testing needs. The automated freeware framework has an integrated setup of various tools serving different purposes like load generation and scripting, resource monitoring, web-page diagnostics, sniffing, database profiling, results analysis and reporting generation.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 11, November 2018

II. RELATED WORK

Milad Hanna et al. [1] have proposed software automated testing framework which can be successfully used for automating the test scripts generation process. This proposed framework is more beneficial when the software under test is changing very frequently and which requires regression testing to confirm the stability of the software version. This framework helped in streamlining the overall software testing automation process by shortening the product launch cycle. It also helped in covering up those manual test cases which was difficult to cover with the traditional test process. The framework also overcomes the limitations of traditional automation techniques by extending the automation setup or activity to involve all the testing tasks. The overall software automated testing framework performance was evaluated in terms of the script creation time, usability, reusability, maintainability and extensibility. The proposed framework saves around 75% of the time or effort involved in automation process using traditional automation methods or techniques.

Nirmala D and LathaMaheswari [2] presented an automated software testing framework which generates the test cases automatically and evaluates it and produces an automated test summary report. This framework can be applied to functional as well as few non-functional testing types. The proposed automated test framework is based on the three models-Structural Test Analysis Model, Behavioural Model and User Interface Model. It is a constructive blend of various strategies, programming standards, methods, perceptions, conventions, system hierarchies, modularity, coverage mechanism and test data injections. This framework helps to organize the test suites and in turn helps to improve the efficiency of testing. It also helps in eliminating the duplication of test cases which is automated across the application. Also, this framework is responsible for specifying the pattern to articulate expectations, building a method to drive the application under test, perform the tests and to testify the results.

Munib Ahmad et al. [3] have presented a proposal for a novel software testing framework to perform class level test. The proposal involves a technique to generate test oracle for specification-based software testing using Vienna Development Method(VDM++) formal language. It describes a three-stage translation process of VDM++ specifications of container classes to C++ test oracle classes. It presents how a derived test oracle is integrated into a proposed functional testing framework. This technique caters the object-oriented features such as inheritance and aggregation. It does not consider the concurrency feature. The proposal also discusses about the Translation issues, its limitations and evaluation of the technique.The test oracle generated using this technique can also be used in parallel with implementation under test to compare the actual and expected results.

C Rankin [4] created STAF (Software Testing Automation Framework) to improve the efficiency and effectiveness of the testing process. This helped in solving the reuse and automation problems. Considerable savings were generated with respect to people, time and hardware necessary to perform the testing activity with the use of this framework. The generated Software Testing Automation Framework was adopted by various test groups across IBM and was used in creating a variety of innovative testing solutions. A two-phased approach was adopted in designing this Automation Framework. The first phase addressed the issue of reuse and the second phase addressed the problem of automation. By providing the reusable framework, STAF allowed the teams to focus on directly solving their problems instead of inventing infrastructure.

Cornel Barna et al. [5] presented a method for performance testing of transactional systems. The presented method models the system under test. It finds the software and hardware performance bottlenecks and generates the workloads which saturate them. This autonomic framework helps in determining the model and workloads during the performance test executions by measuring the system performance. A two-layer queuing model is used using analytical techniques to find the workload mixes which change the bottlenecks in the system. The workloads here are characterized by workload intensity and by the workload mix. The stress vectors that yield a bottleneck change are computed by extracting the switching points from the model. A hill-climbing strategy is later applied for the workload intensity along the stress vectors.

Milad Hanna et al. [6] have presented the main features of different software automation testing frameworks. The Study has highlighted the importance of maintenance of software applications which requires identification and resolution of defects, addition of new features and enhancement of these new features to the existing ones. This requires a lot of regression testing, which consumes a lot of time and efforts. A review on these has emphasized on the use of programmable software automation testing approach. It further highlights two main automation testing approaches which are used in this framework- Record/Playback automation testing framework and Programmable

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 11, November 2018

Automation testing framework. The Record/Playback automation testing framework does not require any advanced testing skills or any programming skills. Whereas, the Programmable automation testing framework requires elevated level programming skills.

Anju Jain M et al. [7] have proposed a web-based cross-platform, code driven object oriented remotely executable software testing framework which is termed as GUI-WAT framework. This framework reads HTML source and generates GUI Web Objects. Selenium tool is used in this framework for providing action-events. Basic modules of the WAT framework include- WebObjects, JSoupParser, WebOperation, Configuration File and Client. This framework seems to be time efficient framework to improve quality and accuracy of testing for the GUI applications. The advantageous factor of this framework is the adaptability of WAT framework with change in GUI application from one version to another version. Another important benefit is the very little human intervention while working with this framework, especially while starting the test-suite and at the end for analysis of results.

F I Vokolos and E J Weyuker [8] have presented an approach to software performance testing of software systems. A case study has been presented describing the experience of applying this approach for testing the performance of the software system which is used as a gateway in industrial client/server transaction processing application. The performance testing objectives were first defined and performance test cases were designed. The Software architecture of the system was then used as a way of identifying the parameters that affected the performance of the system directly. Usage scenarios were also defined by assigning realistic values to the parameters that most directly affected the performance of the system.

Chen S et al. [9] have presented a general purpose testing framework simple, small, complicated and scale performance testing. This proposed framework facilitates performance testing of software applications by separating the application logic from the common performance testing functionalities. This is prototyped on .NET and Java platforms and used for a number of performance related projects. The proposed framework is simple to use and flexible enough to test the performance of complicated and large-scale enterprise systems. This framework also provides a number of test harness designs to facilitate the common functionalities of performance testing of software applications.

Yogita M R and Sangeeta N [10] have presented a reactive based framework for performance testing of Software web applications. This framework approach retrieves web logs from server side with the help of which user patterns are retrieved at the server side. The reactive based framework proposes test case generation in four phases- Web generation and processing, Deriving usage patterns from web logs, Metrics based on Users perspective and through Automated Test case generation. The two metrics derived from Users perspective are PSL value and give up rate. With the help of these two metrics, Usage pattern is derived at client side. These are the inputs to the automated test case generation model. This framework addresses problems like- Metric based problems, issues related to reactivity and access problems in web distributed set up.

Chia Hung K et al. [11] have introduced a Performance testing framework for the REST-based Web applications. This framework provides with an integrated process from test case design, test scripts generation to test execution. Based on the test cases designed and the software artifacts preserved (API document), the framework generates corresponding performance test scripts, which are executed by the performance testing tools. This helps the testers to concentrate more in the design of performance test cases. This framework thus, minimises the effort needed to understand the design and implementation of the application and to learn the operation of the performance testing tools. This way, the efficiency of performance testing increases.

Mahnaz S et al. [12] have proposed a model-based approach for testing the performance of web applications. The generation of synthetic workloads used in the performance tests are simplified in this approach and a formal model to capture application's inter-requests and data dependencies is used. In this model less effort is needed for developing and maintaining the workload generation tools and as compared with the traditional performance testing approach, this model needs very less effort for carrying out the performance testing of software web applications. This approach uses a modified version of Software based web application tester (SWAT) tool. This model consists of a Sequence generator, Trace generator and a Request generator.

J Shaw [13] has presented a study on the performance testing of web applications. According to the study, performance testing can help identifying performance bottlenecks or issues if it is started early in the software development life cycle. It also says that a close relationship should be developed between development and testing teams. The case study says that, just by specifying high performance hardware or some particular software technology

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 11, November 2018

does not result in good response times. The study has concluded with the point that powerful machines do not guarantee good response. The worst performance was recorded in the more powerful environment and the use of particular advanced architecture does not guarantee for adequate performance.

III. PROPOSED FREEWARE FRAMEWORK

Fig.1. shows the proposed freeware framework for Performance testing of software web applications.The framework drives through different components or modules. The Load test script generation module, Bulk load generator or emulator, Web Page diagnostics module, Database Profiling module, Resource Monitoring module, Network Virtualization and Load Test Analysis Module.

Different tools are used at each component or module level. The tester has to decide upon the tool to be used in the framework. A through tool evaluation has to be done before selecting the correct open source tool. Tool evaluation can be based on the following criteria- Scripting ease and compatibility, Load Scenarios configuration facility, Protocol Support and compatibility, Technology support, Resource monitoring capabilities, Data handling ease, Database diagnostics and profiling features, Third party integration facility, Load Test results and statistics collation and analysis features and availability of open source tool experts.

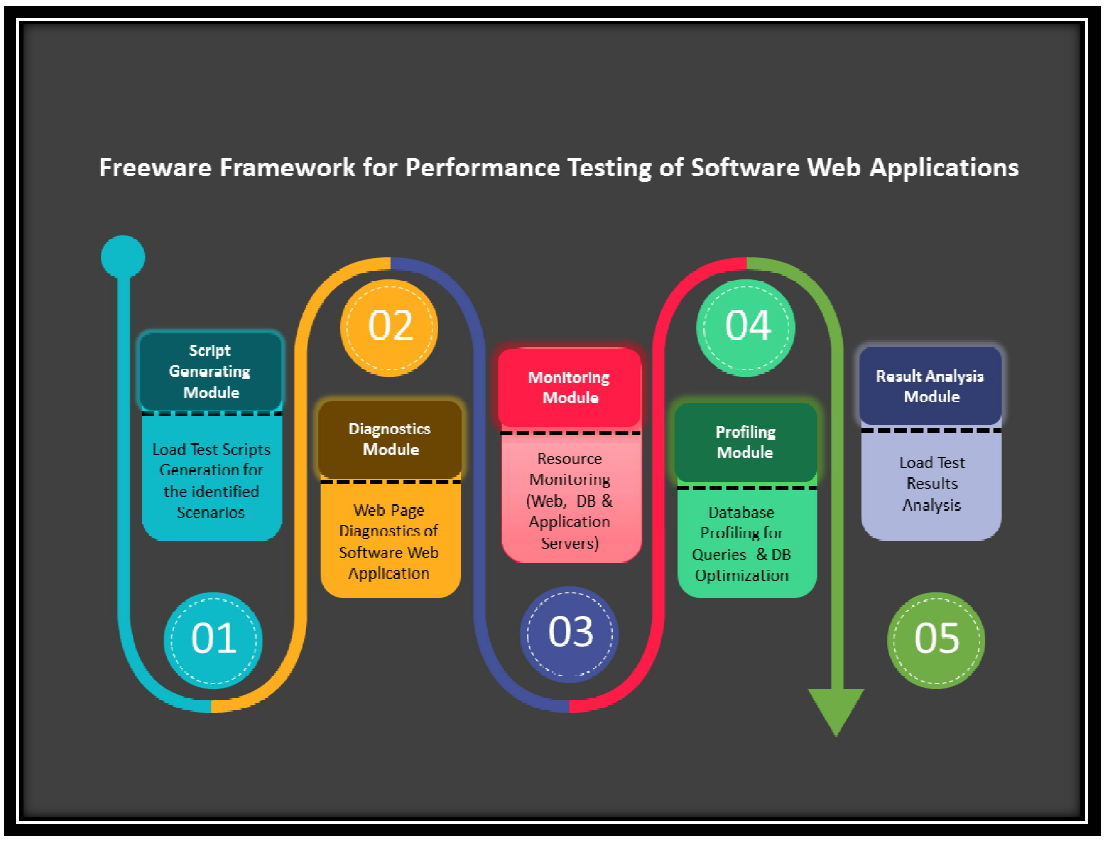


Fig.1. Freeware Framework for Performance Testing of Software Web Applications.

(1) Load Test Script Generation Module:

Load Test scenarios identified during the planning phase is converted into automated scriptsusing the Load test tools. Freeware tools like Apache Jmeter, Blazemeter, Grinder, Locust, Gatling and few more. Grinder and Locust uses Python language and requires expertise tester in these. Grinder supports HTTP, SOAP, JDBC,

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 11, November 2018

SMTP, LDAP and JMS protocols. Apache Jmeter is a Highly Extensible tool with visualization plugins support. It has scriptable samplers which is compatible with languages like Groovy and BeanShell. Multi-protocol support feature is also available in Jmeter. It supports multiple protocols like, HTTP, JDBC, LDAP, JMS, FTP, SOAP, TCP, etc.

(2) *Web Page Diagnostics Module:*
Tools like WebPageTest, YSlow, Lighthouse, Chrome Dev Tools and Web Page Analyser etc. can be used in this module for web page diagnostics. Web Page level metrics like Load Time, Speed Index, First Time, Render time, DOM elements, DNS Lookup, Network round trips are captured and analysed. This module thus helps in optimizing the Web page elements with the diagnostics and drill-down statistics, graphs and reports. The Performance test specialist has to identify the correct and appropriate tool during the tool evaluation process for the Web Page Diagnostics as well. The web pages can thus be optimized with the help of these diagnostics activities.

(3) *Resource Monitoring Module:*
Server's system resources like Memory consumption, CPU usage, Disk usage, Process, Network, I/O are monitored here in this module using different open source monitoring tools. Open Source monitoring tools like, OpenNMS, Nagios, Icinga, Zabbix, Cacti and few more. Cacti can be used for Network monitoring. Apart from these, Perfmon which comes with Window can also be used for windows web application monitoring. Tracers for the monitors has to be enabled and different counters related to these monitors has to be added to record and trace the statistics of the servers. Web, Application and Database servers can be monitored using these tools. This module thus helps in checking the server health during the load test executions.

(4) *Database Profiling Module:*
Database profiling helps in analysing the Databases for Errors, exceptions and performance bottlenecks. The database related performance bottlenecks like deadlocks, Slow running queries, connection pool issues, Cache Issues, wait time issues and etc. are identified using different database profiling and tracing tools at this module. Open Source tools like Neor Profile SQL, dbForge Event Profiler for SQL Server, Express Profiler, IdealSQL Tracer and few more can be used for this purpose. Performance Test Specialist, after doing a thorough evaluation and analysis chooses the correct tool for this module taking into consideration the type of database and the environment. Identified tool is installed or integrated within this framework. After integration, the tracers are enabled, which captures the statistics required for a deep drill-down analysis for optimizing the queries and database. Apart from this tool usage, a server side tracer can also be setup to capture the required events and columns. Thus, this module helps in capturing the metrics related to database queries and databases itself.

(5) *Result analysisModule:*
Load Test result analysis is an important activity in the Performance testing life cycle. All the statistics collected as output from different tools (from other modules) has to be analysed for exceptions, errors, thresholds, SLA violations and other KPI's. Various Open Source tools are used in this module to analyse the captured results. Tools like PAL (Performance Analysis of Logs), Grafana and various other free tools can be used for this task at this module. PAL from Codeplex can also be used for analysing performance counter logs for thresholds. It creates an HTML based report which can be easily copy/pasted into other applications. It is an easy to use GUI interface tool which makes the job easy with the wizard.

IV. CASE STUDY

Study was conducted on Software application developed on .NET and Java Platform. Technology stack included Web Services (REST), Tomcat & MSSQL. The Business critical scenarios were evaluated for response times.Web-UI page response times were also captured during web page diagnostics activities. Apache Jmeter, an open source tool is used for automated script generation and for generation of virtual users. Transaction Response Times for the identified Business transactions for a user load of 250 users is tabulated at Fig.2. It also gives the Pass and Fail percentage of Transactions with the 90th Percentile and Standard deviation values. The transactions taking more time and violating

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 11, November 2018

the SLA's are filtered out from these statistics and a drill down is done for a deep analysis of the results. Code level drill down is also done to analyse the responses through different application layers.

Proj XXX v8

Proj XXX v8 Transaction Response Times for 250VUsers

Transaction Name	Minimum	Average	Maximum	Std. Deviation	90 Percent	Pass	Fail
01_IMC_LOGIN_01_01	4.197	6.655	8.938	0.884	7.693	94	0
01 IMC LOGOUT 01 03	0.144	0.216	1.4	0.161	0.262	85	7
01 IMC MASTERPR AGENT REQUEST PRT 01 04	0.425	0.609	1.997	0.374	0.581	21	0
01_IMC_MASTERPR_AGENT_REQUEST_PRE_01_02	1.126	1.688	2.586	0.364	2.189	21	0
02_PR_ENTER_BATCHINFO_SHIPMENT_02_02	2.075	2.237	2.968	0.198	2.298	16	0
02_PR_RECEIVE_SHIPMENT_PRT_02_05	0.22	0.28	0.397	0.058	0.38	16	0
02 PR RECEIVE SHIPMENT PRE 02 04	5.265	7.595	9.039	0.868	8.618	16	0
03 PR SUB RETAILER INV TRACKING PRT 03 03	0.508	0.712	1.921	0.297	0.817	19	0
03_PR_SUB_RETAILER_INV_TRACKING_PRE_03_02	0.367	0.411	0.672	0.074	0.561	19	0
03_PR_SUB_RETAILER_LOGIN_03_01	0.417	4.586	8.727	2.61	7.282	71	0
03_PR_SUB_RETAILER_LOGOUT_03_06	0.088	0.133	0.688	0.073	0.159	64	0
03_PR_SUB_RETAILER_RECEIVE_ITEMS_PRT_REQ_03_09	1.012	1.173	1.76	0.169	1.352	19	0
03_PR_SUB_RETAILER_RECEIVE_ITEMS_PRE_REQ_03_08	2.121	2.685	3.327	0.358	3.158	19	0
03_PR_SUB_RETAILER_REQ_ITEMS_PRT_03_07	2.944	3.672	4.764	0.503	4.569	19	0
03_PR_SUB_RETAILER_REQ_ITEMS_PRE_03_05	0.654	0.767	1.05	0.11	1.026	19	0
03_PR_SUB_RETAILER_VIEW_HISTORY_03_04	0.249	0.43	2.659	0.527	0.396	19	0
04_MASTER_PR_RETAILER_INV_TRACKING_PRT_04_03	0.452	0.611	1.035	0.141	0.873	19	0
04_MASTER_PR_RETAILER_INV_TRACKING_PRE_04_02	0.349	0.473	1.811	0.338	0.915	19	0
04 MASTER PR RETAILER LOGIN 04 01	7.284	8.5	9.733	0.702	9.515	19	0
04_MASTER_PR_RETAILER_LOGOUT_04_06	0.08	0.098	0.123	0.012	0.118	19	0
04_MASTER_PR_RETAILER_RECEIVE_ITEMS_PRT_04_09	0.99	1.169	1.717	0.174	1.396	19	0
04_MASTER_PR_RETAILER_RECEIVE_ITEMS_PRE_04_08	1.901	2.541	3.627	0.406	3.096	19	0
04_MASTER_PR_RETAILER_REQ_ITEMS_PRT_04_07	0.297	0.367	0.648	0.087	0.565	19	0
04_MASTER_PR_RETAILER_REQ_ITEMS_PRE_04_05	5.034	6.943	8.784	0.999	8.361	19	0
04_MASTER_PR_RETAILER_VIEW_HISTORY_04_04	0.228	0.249	0.282	0.016	0.273	19	0

Fig.2. Transaction Response Times for identified Business transactions.

Tracers are enabled for the database server while the Load test execution is done. Fig.3. gives the list of database queries with their execution time, CPU consumption and Reads and Writes. Database logs also give details on the Exceptions, Errors, Deadlocks, Connections & Cache related and other issues.Neor SQL Profiler was used in capturing the query statistics. These statistics and log data are very much helpful in optimizing and fine-tuning the database layer.

Project XXX

XXX DB Queries -Test1

QUERY	CPU	Reads	Writes	Duration (Micro Seconds)	SPID	START TIME	END TIME
exec [Projxxx].[USP_TRANSACATIONINFORMATION_SEARCH] @IN_DATETYPE=N'TRANSACTION',@IN_INBOUNI	6812	1742505	0	6920437	73	16:04.	16:10.
exec [Projxxx].[USP_TRANSACATIONINFORMATION_SEARCH] @IN_DATETYPE=N'TRANSACTION',@IN_INBOUNI	6610	1742499	1	6657840	51	15:43.	15:50.
exec [Projxxx].[USP_TRANSACATIONINFORMATION_SEARCH] @IN_DATETYPE=N'TRANSACTION',@IN_INBOUNI	6125	1742219	0	6529290	77	28:10.	28:16.
exec [Projxxx].[USP_TRANSACATIONINFORMATION_SEARCH] @IN_DATETYPE=N'TRANSACTION',@IN_INBOUNI	6484	1742491	2	6475127	66	25:22.	25:29.
exec [Projxxx].[USP_TRANSACATIONINFORMATION_SEARCH] @IN_DATETYPE=N'TRANSACTION',@IN_INBOUNI	6109	1628056	3	6317920	67	13:32.	13:38.
exec [Projxxx].[USP_TRANSACATIONINFORMATION_SEARCH] @IN_DATETYPE=N'TRANSACTION',@IN_INBOUNI	6296	1742207	0	6316436	67	18:36.	18:42.
exec [Projxxx].[USP_TRANSACATIONINFORMATION_SEARCH] @IN_DATETYPE=N'TRANSACTION',@IN_INBOUNI	6203	1742461	0	6254195	77	25:34.	25:40.
exec [Projxxx].[USP_TRANSACATIONINFORMATION_SEARCH] @IN_DATETYPE=N'TRANSACTION',@IN_INBOUNI	6078	1742199	0	6227517	77	28:02.	28:08.
exec [Projxxx].[USP_TRANSACATIONINFORMATION_SEARCH] @IN_DATETYPE=N'TRANSACTION',@IN_INBOUNI	5594	1628074	1	6195456	67	21:10.	21:16.
exec [Projxxx].[USP_TRANSACATIONINFORMATION_SEARCH] @IN_DATETYPE=N'TRANSACTION',@IN_INBOUNI	6141	1627944	0	6161911	69	23:26.	23:33.
exec [Projxxx].[USP_TRANSACATIONINFORMATION_SEARCH] @IN_DATETYPE=N'TRANSACTION',@IN_INBOUNI	6016	1627920	3	6046109	67	13:22.	13:29.
exec [Projxxx].[USP_TRANSACATIONINFORMATION_SEARCH] @IN_DATETYPE=N'TRANSACTION',@IN_INBOUNI	5922	1627846	0	6037364	58	21:02.	21:08.
exec [Projxxx].[USP_TRANSACATIONINFORMATION_SEARCH] @IN_DATETYPE=N'TRANSACTION',@IN_INBOUNI	5672	1742477	1	5989374	88	15:55.	16:01.
exec [Projxxx].[USP_TRANSACATIONINFORMATION_SEARCH] @IN_DATETYPE=N'TRANSACTION',@IN_INBOUNI	5828	1742484	0	5910831	65	25:42.	25:48.
exec [Projxxx].[USP_TRANSACATIONINFORMATION_SEARCH] @IN_DATETYPE=N'TRANSACTION',@IN_INBOUNI	5828	1742224	3	5863273	80	27:51.	27:57.
exec [Projxxx].[USP_TRANSACATIONINFORMATION_SEARCH] @IN_DATETYPE=N'TRANSACTION',@IN_INBOUNI	5579	1742700	8	5843451	57	18:18.	18:23.
exec [Projxxx].[USP_TRANSACATIONINFORMATION_SEARCH] @IN_DATETYPE=N'TRANSACTION',@IN_INBOUNI	5703	1627516	0	5803601	78	30:34.	30:40.
exec FINANCE.USP_SPECIALJOURNAL_GET @IN_SPJID=0,@IN_COAID=0,@IN_STARTDATE='2014-01-01 00:00:00',	1297	69655	657	5766907	64	21:05.	21:11.
exec [Projxxx].[USP_TRANSACATIONINFORMATION_SEARCH] @IN_DATETYPE=N'TRANSACTION',@IN_INBOUNI	5704	1627022	0	5766159	69	23:19.	23:25.
exec [Projxxx].[USP_TRANSACATIONINFORMATION_SEARCH] @IN_DATETYPE=N'TRANSACTION',@IN_INBOUNI	5750	1628445	3	5764516	67	13:11.	13:17.
exec [Projxxx].[USP_TRANSACATIONINFORMATION_SEARCH] @IN_DATETYPE=N'TRANSACTION',@IN_INBOUNI	5297	1627494	0	5447979	76	30:27.	30:32.
exec [Projxxx].[USP_TRANSACATIONINFORMATION_SEARCH] @IN_DATETYPE=N'TRANSACTION',@IN_INBOUNI	5391	1742205	0	5435812	67	18:28.	18:34.
exec [Projxxx].[USP_TRANSACATIONINFORMATION_SEARCH] @IN_DATETYPE=N'TRANSACTION',@IN_INBOUNI	5063	1627926	4	5102637	75	23:08.	23:13.
exec [Projxxx].[USP_TRANSACATIONINFORMATION_SEARCH] @IN_DATETYPE=N'TRANSACTION',@IN_INBOUNI	5000	1627500	3	5082470	81	30:16.	30:22.

Fig.3. Database query traces captured while profiling during the execution of Load test.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

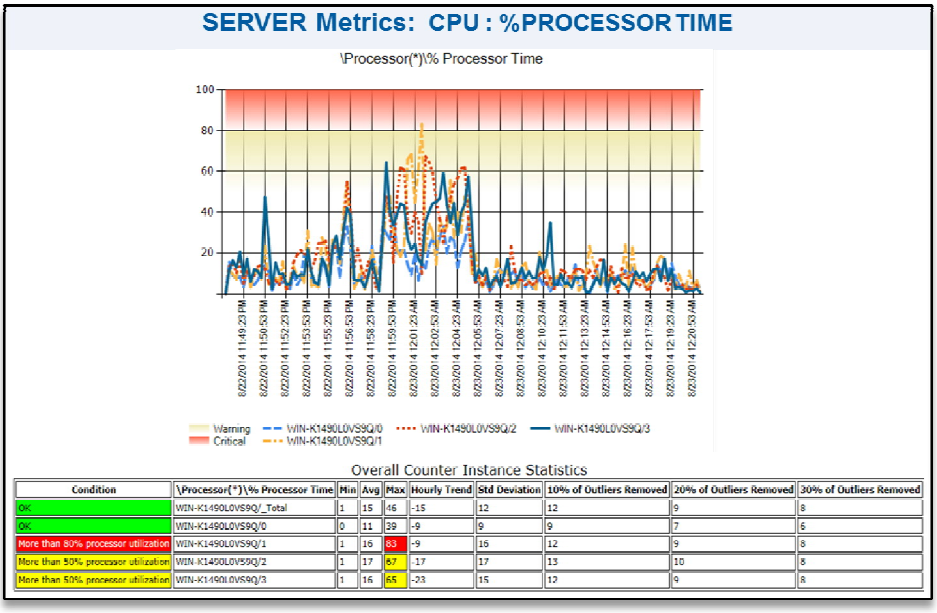
International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 11, November 2018

Perfmon and OpenNMS were used to capture server statistics during the Load test execution. Monitoring tracers were set up for the Database, Web and Application servers. Required counters for the Disk, Process, Network and Memory monitors were added for these servers. Fig.4. gives the CPU % statistics for the server. These tabulated data with graphs are obtained from the PAL(Performance Analysis of Tools) tool. Warnings, Violations and general metrics are highlighted in Red, Yellow and Green.



Condition	\Processor(*)\% Processor Time	Min	Avg	Max	Hourly Trend	Std Deviation	10% of Outliers Removed	20% of Outliers Removed	30% of Outliers Removed
OK	WIN-K1490L0VS9Q/_Total	1	15	46	-15	12	12	9	8
OK	WIN-K1490L0VS9Q/0	0	11	39	-9	9	9	7	6
More than 80% processor utilization	WIN-K1490L0VS9Q/1	1	16	83	-9	16	12	9	8
More than 50% processor utilization	WIN-K1490L0VS9Q/2	1	17	67	-17	17	13	10	8
More than 50% processor utilization	WIN-K1490L0VS9Q/3	1	16	65	-23	15	12	9	8

Fig.4. Server Statistics obtained are analysed through PAL(Performance Analysis of Logs) tool.

Database server metrics were also captured for studying the server health.Fig.5. shows the data obtained from PAL tool for the SQL Server used in the testing. The areas of concern were easily identified using this metrics. The database related Warnings and Critical areas highlighted in Yellow and Red are analysed for the root cause. Based on the drill-down analysis and root cause analysis, the areas of concern are addressed and fine-tuned. Thus, the metrics obtained from Monitoring module are very much helpful in optimizing the servers and software application.

DB Server Metrics

MONITOR	CONDITION	PERFORMANCE COUNTER	MIN	AVG	MAX
SQL Server	Greater Than > 1	SQLServer:Locks(_Total)\Lock Timeouts/sec	0	101,852	311,890
	Greater Than > 1	SQLServer:Locks(Key)\Lock Timeouts/sec	0	99,789	305,523
	Greater Than > 1	SQLServer:Locks(Page)\Lock Timeouts/sec	0	2,063	6,367
	Greater than 0	SQLServer:Locks(*)\Lock Waits/sec/_Total	0	0	1
	Lock Requests/sec Greater than 1000	SQLServer:Locks Lock Requests/sec/Total	3	240,911	1,712,390
Physical Disk	More than 2 I/O's are waiting on the physical disk	\PhysicalDisk Avg. Disk Queue Length/ 0 C:	0	356	2,472
Logical Disk	More than 2 I/O's are waiting on the logical disk	\LogicalDisk(*)\Avg. Disk Queue Length/ C:	0	356	2,472
Processor	More than 80% processor utilization	Processor(*)\% Processor Time\WIN-K1490L0VS9Q/1	1	16	83
	More than 50% processor utilization	Processor(*)\% Processor Time\WIN-K1490L0VS9Q/2	1	17	67
	More than 50% processor utilization	Processor(*)\% Processor Time\WIN-K1490L0VS9Q/3	1	16	65
Memory	More than 70% the Commit Limit is in use	\Memory\% Committed Bytes In Use	79	80	82

Fig.5. Database server (SQL Server) analysis through PAL tool.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 11, November 2018

V. CONCLUSION

The proposed Freeware Performance Testing framework can be successfully used for various software web applications and products for the identification of performance bottlenecks and optimization. This framework can thus cater the performance testing needs and can very well fit in the Performance testing life cycle. This framework was effectively used in testing various software web applications and the results were quiet promising as this was very much helpful in identifying the performance bottlenecks. Both, Client-side and Server-side bottlenecks were successfully identified and fine-tuned using this framework setup.Different types of performance tests like, Load, Stress & Stability or Endurance can be conducted using this framework.As, this framework uses all the open source or community or trial version software, it saves a lot of time and money. It is thus a cost-effective and Business effectiveframework and can give a High ROI (Return on Investment).

REFERENCES

[1] Milad Hanna, AmalElsayed A and Mostafa-Sami M, "Automated Software Testing Framework for Web Applications", International Journal of Applied Engineering Research, Vol.13, No. 11, pp. 9758-9767, 2018.
[2] Nirmala D and LathaMaheswari T, "Automated Test Framework for Software Quality Assurance", International Journal of Computer Science and Mobile Computing", Vol.4, Issue 12, pp. 224-234, 2015.
[3] Munib Ahmad, FuadBajaber and RizwanJameelQureshi, "The Proposal of a Novel Software Testing Framework", Life Science Journal, Vol.10, No. 4, pp. 319-326, 2013.
[4] C Rankin, "The Software Testing Automation Framework", IBM Systems Journal, Vol. 41, Issue 1, pp. 126-139, 2002.
[5] Cornel Barna, M Litoiu and H Ghanbari, "Autonomic Load-Testing Framework", Proceedings of the 8th ACM international conference on Autonomic computing, pp. 91-100, 2011.
[6] Milad Hanna, AmalElsayed A and Mostafa-Sami M, "Automated Software Testing Frameworks: A Review", International Journal of Computer Applications, Vol. 179, No. 64, pp. 22-28, 2018.
[7] Anuja Jain M, Swarnalatha P and S Prabhu, "Web-Based Automation Testing Framework", International Journal of Computer Applications, Vol. 45, No. 16, 2012.
[8] F I Vokolos and E J Weyuker, "Experience with Performance Testing of Software Systems: Issues, an Approach, and Case Study", IEEE Transactions on Software Engineering, Vol. 26, No. 12, pp. 1147-1156, 2000.
[9] Chen S, Moreland D, Nepal S and Zic J, "Yet Another Performance Testing Framework", In: Australian Conference on Software Engineering (ASWEC), pp. 170-179, 2008.
[10] Yogita M R and Sangeeta N, "Web Application: Performance Testing Using Reactive Based Framework", International Journal of Research in Computer and Communication Technology, Vol. 4, Issue 2, pp. 114-118, 2015.
[11] Chia Hung K, Chun Cheng L and Juei-Nan C, "Performance Testing Framework for REST-Based Web applications", Proceedings of 13th International Conference on Software Quality, pp. 348-353, 2013.
[12] Mahnaz S, Diwakar K and Behrouz F, "A Model-based approach for Testing the Performance of Web Applications", Proceedings of the Third International Workshop on Software Quality Assurance, pp. 54-61, 2006.
[13] J Shaw, "Web Application Performance Testing- a Case study of an Online Learning application", BT Technology Journal, Vol. 18, No. 2, pp. 79-86, 2000.