Quality Assurance and Quality Control: The concept of performance testing

Pankaj N Moharil[1], Sudarson Jena[2], V.M.Thakare[3]
[1]Research Scholar, SGBAU, Amravati, Maharashtra, India
[2]Department of CSE & A, SUIIT, Sambalpur, Odisha, India
[3]Department of CS & E, SGBAU, Amravati, Maharashtra, India

Abstract- In the present competitiveness commercial center business looking for most lucrative worthwhile results need to guarantee that the systems they are utilizing stand healthy so as to pass on persistent application accessibility. For business development, this is essential for ventures as the elite of business applications. Consequently the most significant for system managers is to keep track of the clean of their servers with the goal that the rigorous applications which are suffering a result of responsive strategy over watching servers and applications in the development stage.

Hence the performance problems can be inclined only after they have occurred, regularly affecting operations and business. To address these challenge organizations need performance testing to identify issues before the production stage in the systems development life cycle. Anyway, effective performance testing requires to be performed consistently and ought to consider changes in workload, validity, and performance inferable from business development and advances. Development organizations require a positive methodology that can convey an auspicious and comprehensive perspective on an application's performance in this manner relieving cases of application failures and beating performance problems and server down instances. This paper describes quality assurance, quality control, and performance testing methodology that empowers organizations to help from predictive and continuous performance and workload estimation of their applications and infrastructure.

Keywords – Performance, Quality assurance, Quality control, Performance testing

I. INTRODUCTION

In this digital era, the performance of IT systems has a crucial role to play in determining the accomplishment of an organization. The development teams need to perform innumerable activities to observer the system performance/ workload defeating business troubles or slow performance issues and system setup updates. In such a situation performance testing is crucial to assure that application critical to ensure that application end sure does not face performance problems in the development phase. Performance testing activities are developing from receptive to progressively comprehensive and concentrating on the quality of the software using proactive methodologies. For successful software performance, testing business requires to consider the aspects of software quality, workload and capacity management issues. Performance measurement of all factors should be done in a respective manner. This can be accomplished by using performance testing techniques with continuous capacity management procedures in an all-encompassing way [1].

While purchasing any product from the market usually the customer is curious about its quality and assurances. For example, when you purchase fruit swiftly you can evaluate its quality and performance: the size, shape, taste, etc. but only as you take the first bite of fruit, you will be able to identify that the fruit you are buying is really good or not. Sometimes good looking fruits might be tasteless or not healthy. The same things relate to all products, be it a software application. Web applications you find on the internet might look good, but as you start surfing through that web application it may show the errors in linking pages, in design flaws and performance issues.

An e-commerce web application that has performance problems might cost the stakeholders amounts in revenue. That's why the paper discusses quality assurance, quality control, and performance testing process, best practices and ideal approaches [2].

Software performance engineering: Software performance engineering identifies the issues in the early development and uses qualitative and quantitative approaches to help cost-effective analysis, workload estimation and software capacity administration. Software performance engineering is a systematic method for building software systems to encounter performance goals. It is too costly to build reactive software applications for that the development team should use performance testing and its new methods early in the software lifecycle [3]. The development team should concentrate on the quantitative approaches, improvements in development and maintenance productivity; the focus should be on identification of problems before implementation, use of new approaches, tools, and models for performance testing. They might use through analysis of requirements, detailed design, coding and testing for preventing issues such as delay delivery and tricky code maintenance issues. Some developers may think of tuning the software applications after development for improving the performance of the application. Though tuning may improve performance, but not as much as design can. Performance problems are usually caused because of fundamental design and architectural problems; it's regularly very costly and infeasible to make changes in the fundamental design selections [4]. Tuning might present tricky code to resolve the problems but it may the main alternative for accomplishing desired objectives late in the development lifecycle. Traditionally a developer uses common approaches where functional requirements are considered only during the development & identifies performance requirements at the end.

II. PERFORMANCE TESTING

Performance testing is an important attribute of the software development cycle. As it is important to attribute of software development cycle question arises why it is performed? Only one reason to perform software performance testing is to provide value to the stakeholder. The essential goal of performance testing is to determine how a system can respond to a specific workload or task and checking for the software standards are met, and benchmarks are attained. Performance testing helps in determining various issues like software design, Server configuration, and hardware limitations. Performance testing actuates capability and usability of the software and measures validates and verifies the quality attributes of the system such as scalability, responsiveness, stability, speed under a variety of load conditions and varying workloads and it also focuses on system restraint, user expectations and cost. Performance testing assists in identifying the performance and responsiveness of applications and also helps in business to recognize the efficiency and effectiveness of the targeted goal of the product in real-time [5]. While conducting performance testing of software some points should be considered as

- Prepare sensible performance objectives
- Use of successful performance testing technique
- Clarify performance test results
- Endure with various application technologies and structures
- Perform testing using automated testing tools
- Test traditional local applications, web-based applications, and web services (SOA)

Recognize and resolves issues that are regularly ignored in Performance tests Prioritization of performance testing objectives fundamentally increases the value of the performance testing effort. Tester team must identify that for whom and for what purpose performance testing is going to be performed [6]. Tester team should identify the business stakeholders, end-user and regulatory inspections objectives for successful performance testing

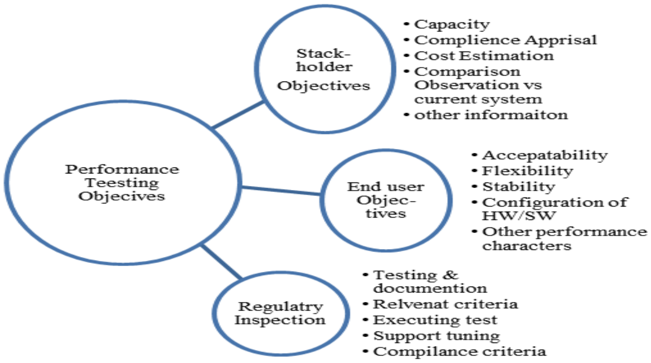

Figure 1 Objectives of software performance testing

Objectives that support business stakeholders for performance testing include the following:

- Estimates capacity and scalability
- Define observations vs. current systems
- Standards of user gratification
- Likelihood of costly or humiliating disappointments (with alleviation proposals)
- Compliance appraisals
- Other data that enhances go-live choices

Objectives for performance testing as a point of view of end-users include the following:

- Appraisal of acceptability and flexibility of response times independent of load
- Appraisal of stability and functional integrity independent of load

- Assessment of performance worthiness of prescribed customer Hardware/software/designs
- Assessment of other performance characteristics likely to reduce quality in the eyes of the end-user.

Objectives that support performance testing objectives for a regulatory inspection:

- Testing and documenting in compliance with relevant standards
- Determine compliance with relevant criteria
- Building and executing tests that replicate regulatory/compliance tests

III. QUALITY ASSURANCE AND QUALITY CONTROL

First, the basic question arises about the quality of the software and its Performance. The concept of software quality is that the software is secure and functioning as expected, where the functional and non-functional requirements meet the specified goals [7]. The operational quality of the software application is regularly difficult to achieve: it depends on the expert team of engineers and maybe assured through analysis and refactoring. The functional characteristics may be assured through strict quality management actions such as quality assurance, quality control, and performance testing. Performance testing involves a number of qualitative and quantitative tests performed while gathering requirements, measuring scalability, availability and response time of the system. Software quality is always difficult to characterize, it is expressed as the software fulfilling the client's needs and meeting the desired goals of the client's concerning all functionality, structure, reliability, strength, and cost of the item [8].

3.1 Software quality assurances

It is a positive revelation on software and its services. Quality assurance or quality testing centers around improving the procedures to convey quality software products to the client and it is the conviction of software that it will function admirably and gives assurance that the software will work without any problems. It is characterized as the actions taken to ensure that the company or organization is giving the most ideal or best service to the client. Organization or company needs to guarantee that procedures are effective and proficient according to quality standards for software products. Basically, software quality assurance has a quality management approach with various testing strategies, formal technical reviews, compelling software engineering technology with analysis, measurement and revealing system [9].

3.2 Quality assurance organizations have to build some enactments they are as follows

- Generating a robust test environment and anxiously selecting the released benchmark.
- Use automated tools for testing to the high-risk area to set aside cash
- Design structure for devoted performance testing team and committed security
- Proper time management for each procedure and replicate client accounts like a generation domain
- It is imperative to set up bug fixes dependent on software use.

3.3 Software Quality assurance Process:

Quality assurance is a cyclic process, where planning, determination, analysis and execution stages are used to develop a quality software product. Important software quality assurance activities are managing plan, validation, and evaluation of performance, use of various testing methods and maintaining a healthy environment in the organization.

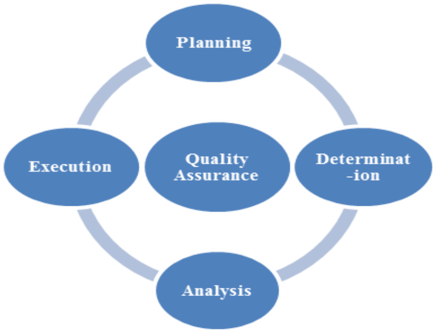

Figure 2 Quality assurance process

Planning: In the planning, stage the organization should design and set up the procedure for related goals and objectives. They should decide on the procedures that are required to convey the best quality product.
Determination: The development should determine the development and testing procedures and furthermore should make the changes in procedures.
Analysis: In this stage development should perform the observation of procedures, change the procedures and analysis of procedures whether it meets the predetermined targets or not.
Execution: Execute activities that are important to accomplish enhancements in the procedures [10].
In the quality assurance process above stages are repeatedly executed to guarantee that procedures used in the organization are assessed and enhanced an occasional premise. An organization should utilize quality assurance to guarantee that software product development is planned and executed with the right methods. Quality assurance produces excellent software & advantageous for better reliability, saves time and money. Advantageous in developing high quality, maintenance-free business software and it improves the quality of the product [11].
Software Quality Control

Quality control is a process where a lot of techniques are utilized by the organization to accomplish quality objectives and consistently improve the software organization's capacity to guarantee that a product will meet its quality objectives. The quality control process is performed to convey a quality product to the customers at its best cost. The main goal is to compare the quality of products with other organizations to gain different organizations' skills of quality that would help to improve to quality of product each time and accurate planning for abstaining from making errors [12]. The basic parameters of software quality control processes are product, procedures, and resources available. The software quality control process is performed with four stages concentrating on a project, process, and resource.

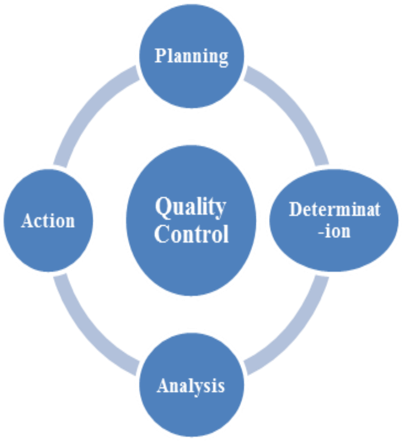

Figure 3 Quality control process

- Planning: In the planning stage total quality control procedures are planned in a proper manner.
- Determination: determines & characterize the quality parameters to build up the quality product.
- Analysis: analysis of the quality parameters to confirm if the nature of the parameters is met.
- Action: stages take remedial activity if necessary and repetition of work.

The quality control process inspects the quality of the end product and the final result. The basic aim is to verify that the developed product meets the specification and objectives of the client. If an issue is identified then it required to be resolved before delivery of the product to the client. Quality control certifies that the prescribed procedures and standards are trailed by the development team to develop a quality software product. Software quality feature is concentrated by measuring software quality metrics. Software quality metrics are software Product quality measurements, In-process quality measurements, Maintenance measurements [13].

- Product quality measurement includes mean time to failure, fault density, client issues and satisfaction.
- In-process quality measurement includes tracing of faults that occur at the testing. They are follows
- Fault density and fault appearance in machine testing
- Effective fault elimination and stage based fault elimination design

- During maintenance measurement fix excess and excess management ratio, ratio offending fixes, fix response time and fix quality such type of fixes are performed to remove the faults for better fix quality.

IV. SOFTWARE PERFROMANCE TESTING PROCESS

Performance testing is led to give subtleties data about the nature of the item or administration under test with a huge load, respect to speed, scalability or potentially steadiness attributes and also identifies bottlenecks of the system to provide good quality of service to users. Performance testing in-depth notices the performance metrics of web servers, application servers, database servers, and other support elements and creates some critical resolution about system architecture; tuning and hosting alternatives that show exact tips that could check the performance and experience application speed. Continuous integration of test, evaluation, and optimization of web applications improves the quality of software

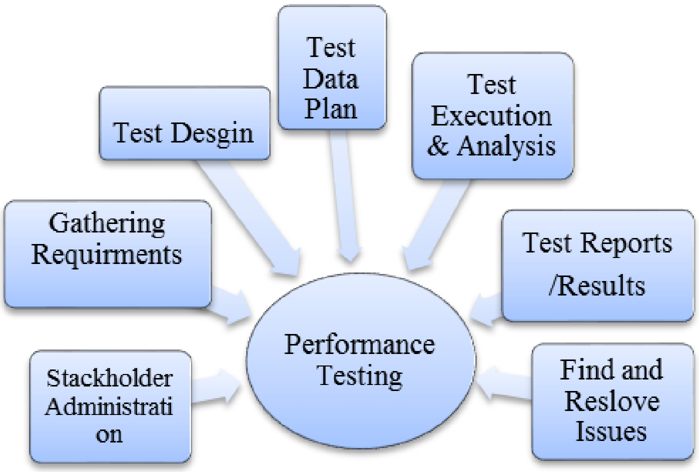

Figure 4 Performance testing process

Performance testing has limitations about user satisfaction, business scope, budget and time limit. Before performance testing development people or tester have to recognize some points regarding web application software operations and its environment. The purpose is to check web applications carefully with functional and nonfunctional factors of web applications in all testing levels, where some other tests of web applications must be performed are a unit test, acceptance test, conformance, and robustness test [14].

- Acceptance Test: determines the applicability of an application to the automation, usability performance and robustness requirements of an application for the satisfaction of users.
- Conformance Test: matches the performance of the application to its stated requirements.
- Robustness Test: it checks over the application's capacity to perform tasks in ways which are not calculated & the applications capability & possibility to be modified and scaled away from the original requirements.

Performance testing is a remarkable assignment where tester must understand what the application does, usage profile and the methods it uses to operate correctly. Web application issues exhibit in numerous ways, along with the atrocious input data, the invalid sequence of events, failures in system resource possibility and response [15]

Web applications at any sort provided would be failed because of bumbling web pages and site presentation, navigation, deficient administration of client-side, network and server-side interface problems & mismanagement of data castling. The tester must be accomplished with the countless interfaces available to the web applications because of tremendous growth in different web browsers, client systems, and network systems. Tester team has to recognize the web application environment to identify the web application types, types of interfaces & interaction with the other elements in the web application’s universe. The essence of these elements creates a model of the application environment in which web application runs or executes [10].

4.1 Performance Measurement

The tester team measures the performance of web applications. Web application performance targets get derived from the relationship between transactions, batch jobs and reports depend on business workload [16]. Habitually customers give a number of simultaneous users and respond to time objectives, yet not output targets. So the Performance target validation becomes an important factor in specifying the performance targets requirements. Through validation of performance requirements system load will be easy to get calculated with the number of users, throughput or output and response time. It is important to know the connection between the number of uses, throughput or output and response time. Network traffic load and release times should be validated.

The Workload estimation includes no functionalities of a software system with the estimated response time. Performance measurement of the web application is basically categories into two categories qualitative and quantitative; where product efficiency and service quality are two major factors concentrated on which web application performance is measured [17]. To do the accurate performance measurement key performance indicators must be considered or introduced with two factors product efficiency and service quality. Performance testing with qualitative measurement specifies the appropriate test cases, run the scenario and then catches the performance measurement. Performance testing with quantitative measurement specifies average workload & estimated time.

4.2 Types of Performance Testing

Performance complications in web applications are highly-priced both in terms of business and alleviative work. While doing the performance testing tester must be aware of the performance characteristics of web applications [18]. Performance characteristic includes load, stress, scalability, reliability and volume Test.

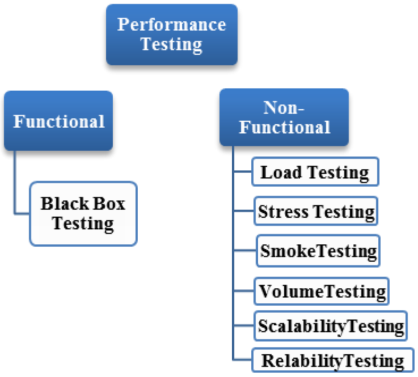

Figure 5 types of performance testing

- Load Test: Load test specifies how web applications responding to the different simulated load.
- Stress Test: This testing specifies web application stability beyond the normal or peak load conditions.
- Scalability Test: Testing specifies the degree at which the application will scale up and scale out in terms of user loads and transactions.
- Reliability Test: This test specifies how long the web application can sustain good response levels under load conditions.
- Volume Test: This Test specifies how web application responses under different volumes of data.
- Smoke Test: In this test, the tester would be able to determine the functioning of a particular system as against the previous versions.
- Component Test: This test specifies whether the component is doing what is expected and whether optimization of the components is performed correctly

4.3 Performance Metrics

It is important to understand the performance metrics while conducting performance testing for better results. The Tester team or developers should establish the performance test fundamental metrics to be collected and concentrated. Performance metrics compared to the performance requirements to decide whether the system is meeting the business need [2]. Performance metrics are expressed as follows:

Response time to each online interaction: Communication time between screens is accomplished as response time

Commonly Response time is calculated in the act of mainstream amount or proportional amount.

Positioned on the type of interaction

Positioned on network bandwidth

Positioned on the overall completion time of the transaction

Completion time to each online transaction: Time to require completing an online transaction. Completion time is comprehensive of the data entry time, time to respond to the activities and time required to collect information from end-users.

Transaction output: Number of transactions processed per unit time accredited to the output.

Batch completion time: Completion time requires by the batch program along with the backups.

Report throughput and completion time: Reports might be evaluated as subsistence to the batch operation and transaction processing. Presented reports have to be classified into anticipated and improvised reports [19]. With every category of the report, the output should be stated as mainstream and apex loads. With each improvised report, the completion time should be stated; however, it should have addicted values to control the submission of ad hoc reports. The improvised reporting system is better than the other reporting systems.

Resource consumption: Resource consumption at each level of an operating system calls with the following factors:

CPU usage time: Percentage of the time required by the CPU of the system when it is active.

Memory utilization: How much memory utilized by the system RAM in the number of MB or GB.

Network bandwidth: Complete Network utilization in terms of Kbps or Mbps.

V. SOFTWARE PERFORMANCE TESTING PROCESS

Performance testing is testing mainly categorized into two categories performance scripting and performance observation

5.1 Performance scripting

Load testing reproduces a genuine load on any application or site. Stress testing decides the security and heartiness of the system. It is performed for making/scripting the workstreams of a key distinguished business process. The performance testing instruments assume a basic job in guaranteeing that the performance of the discharged programming meets the concurred Service Agreement Levels (SLAs) [1]. The performance is the way to offer higher client fulfillment levels. This should be possible utilizing a wide assortment of tools.

Here is a rundown of Top 5 performance testing Tools that can help in discharging elite programming.

WebLOAD: WebLOAD is the tool of decision for endeavors with substantial client load and complex testing prerequisites. It allows you to perform load and stress testing on any web application by producing load from the cloud and on-premises machines [10]. WebLOAD's qualities are its scalability and convenience empowering you to rapidly characterize the tests you require with highlights like DOM-based account/playback, programmed connection, and JavaScript scripting dialect.

The tool gives an unmistakable investigation of your web application performance, analytical issues, and bottlenecks that may hinder accomplishing your load and response prerequisites. WebLOAD supports several innovations from web conventions to big business applications and has worked in the mix with Jenkins, Selenium and numerous different instruments to empower nonstop load testing for DevOps [7].

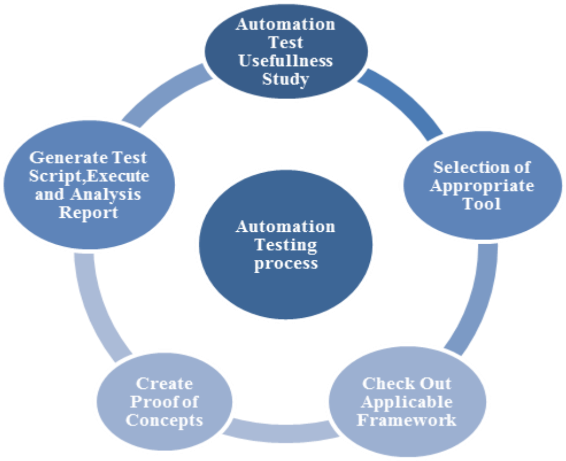
Automation Test Usefullness Study
Selection of Appropriate Tool
Check Out Applicable Framework
Create Proof of Concepts
Generate Test Script,Execute and Analysis Report
Automation Testing process

Figure 6 Performance test automation tool feasibility check process

LoadUI NG Pro: Create modern load tests with only a couple of snaps. LoadUI NG Pro makes it simple to show true loads on your API. LoadUI bolsters REST, SOAP, JMS, MQTT and numerous other API positions. LoadUI's anything but the difficult to-utilize graphical interface makes it basic for new users to set up load situations and furthermore gives progressed scripting highlights to those with more experience [8]. This device permits investing more energy breaking down the outcomes than on designing and building tests by hand. You can adapt much progressively about your application by making utilization of LoadUI's capacity to provide details regarding server performance information. You can set up nearby or all-inclusive appropriated stack operators for your heap testing situations. LoadUI NG Pro effectively enables you to reuse your current useful experiments from SoapUI and SoapUI NG Pro. With only three ticks you can change your SoapUI NG Protest cases into modern performance tests.

Apache JMeter: It is a Java stage application. It is basically considered as a performance testing tool and it can likewise be coordinated with the test plan. Notwithstanding the heap test plan, you can likewise make a useful test plan. This tool has the ability to be stacked into a server or system in order to beware of its performance and examine its working under various conditions [20]. At first, it was presented for testing the web applications, yet later its degree had broadened. It is of incredible use in testing the utilitarian performance of the assets, for example, Servlets, Perl Scripts, and JAVA objects. Need JVM 1.4 or higher to run.

Load Runner: This is an HP item that can be utilized as a performance testing tool. This can be purchased as an HP item from its HP software division. Additionally, it is especially helpful in comprehension and deciding the performance and result of the system when there is a real load [5]. One of the key appealing highlights of this testing tool is that it can make and deal with a large number of users in the meantime. This device empowers you to assemble all the required data regarding the performance and furthermore dependent on the system. The Load Runner contains diverse devices; to be specific, Virtual User Generator, Controller, Load Generator and Analysis.

NeoLoad: This is a device utilized for estimating and investigating the performance of the site. The performance and the final product can be assessed by utilizing this tool and any further advances can be taken. This causes you to enhance and streamlining the performance of your web application. This tool investigates the performance of the web application by expanding the traffic to the site and the performance under substantial load can be resolved [9]. You can become acquainted with the limit of the application and various users it can deal with in the meantime. This tool was produced by a French organization named Netosys and it was written in JAVA.

5.2 The database used in performance testing

The Database is a pivotal component of any software system which lies at the backend to give full help to the application to store and recover information from it. A database application with high response time and poor performance can prompt enormous issues. Database Load Testing supports in identifying the performance problems before deployment database applications for end-users [12]. It helps in the design and simulating usage traffic which can be used to test database application infrastructure for performance, reliability, and scalability. Load testing of database applications involves testing the performance and scalability of Database applications with changing user load. Database performance testing is done with several databases such as Oracle, SQL Server, DB2, PostgreSQL, MySQL, etc. in order to keep the database running smoothly and can actively monitor metrics performance and resource utilization. Here is a list of some Database performance testing tools that can help in releasing high-performance software.

DTM DB Stress: It is software for load testing and stress testing the server parts of data frameworks and database applications, just as databases and servers themselves. It is reasonable for arrangement scalability and performance testing, examination, and tuning. The apparatus enables clients to make and design a constant arrangement of solicitations to the server of the OLAP (question execution) and OLTP (including/loading, changing and erasing information in the database) types [21]. The user can specify the number of virtual users to be matched, priority and type of requests for each job (virtual user type).

DB Fit: DB Fit is an open-source database testing structure that bolsters simple test-driven advancement of your database code. DB Fit is written on top of Fitness, a mature, fully-featured framework with a large community. Tests are composed utilizing tables, making them clearer than x Unit-style tests [22].

Hammer DB: Hammer DB is an open-source database load testing and benchmarking apparatus. It is mechanized, and extensible with dynamic scripting support multi-threaded and extensible with dynamic scripting support. Hammer DB incorporates total implicit workload dependent on industry-standard benchmarks just as catch and replay for the Oracle database [3].

Sqlmap: sqlmap is an open-source infiltration testing tool that mechanizes the way toward recognizing and abusing SQL infusion defects and taking over of database servers. It accompanies an incredible location engine, numerous specialty highlights for a definitive entrance analyzer and a wide scope of changes enduring from database fingerprinting, over information getting from the database to getting to the fundamental file system and executing directions on the working system by means of out-of-band associations.

New approaches introduced in software performance testing
New ways to deal with enhancing product performance is deciphering and improving performance through novel advancement, combination, and parallelization of innovation. The testing group should begin considering life cycle virtualization space, similar to client virtualization, benefit virtualization, and organize virtualization and information virtualization that will rouse the group to accomplish the Performance objectives [23]. Here are some new ways introduced for making successful software performance test as below:
Adoption of DevOps: For effective software performance test analyzer group ought to embrace DevOps. It is a test tool for the software development approach which consolidates software improvement with data innovation activities. Embracing DevOps with the software improvement process will build the product performance rapidly and precisely. With the DevOps, the testing begins toward the start of the development cycle. This improvement approach permits consistent joining and nonstop conveyance [24]. This permits the analyzer group to perform consistent testing and ceaseless observing to approve that the designers have manufactured the correct application. The usefulness and performance of the application are tried persistently alongside the development. The testing group adjusts the test configuration, and experiment developments with DevOps to check the code changes as well as guarantee that the progressions don't break the item [25]. The capacity to apply a ceaseless testing approach as a major aspect of the all-encompassing testing methodology is an extraordinary aptitude to learn. At long last, figures out how to give constant criticism to designers, as a type of theory about changing the code to help better performance following.
Machine Learning and Artificial Intelligence: A crowd of cutting edge improvements is gotten the IT business by Artificial Intelligence. The inside stage is as of now ruled by common dialect preparing, neural systems, and machine learning [4]. The best thing about man-made brainpower is the limit of a framework in growing more systems. Machine learning can be utilized in the saving money space to display constant exchanges and. Machine learning alongside AI can likewise anticipate deceitful exchanges. Associations that are prepared to plunge into the computerized space are going to try different things with high effect situations created by machine learning in the next years. An analyzer must be prepared to comprehend the advancing patterns in the testing space and so as to remain aggressive should create experiment situations that can approve all the conceivable results [11]

VI. CONCLUSION

The main concentration of this paper is on software quality assurance, quality control, and different types of web application & its services performance testing methodologies and how to improve the performance testing of software applications. The purpose is to concentrate on the important aspects of performance testing such workload, scalability, and reliability which are useful in performance evaluation and testing performance metrics and various parameters. In this paper, the focus is on performance characteristics, multiple attributes and parameters of the software performance and aspects of software quality assurance and quality control. Basically, these three attributes are important factors of software performance; adequate performance measurement helps in improving the performance of the software system.

VII. FUTURE SCOPE

The approach presented in this paper investigates performance engineering by looking into various patterns to implement a performance test framework. Another way to improve the performance of the test system is to efficiently balance the load over the computation resources. The problem of efficient allocation of resources or resource scheduling is encountered very often. The complexity of the tested systems is increasing rapidly as well as the demand for higher quality and more features. The performance testing tools for typical client-server architectures require the simulation of thousands of users; this requirement raises the problem of how to design a proper system to sustain the effort of thousands of users by using cost acceptable software. To achieve that, the load balancing mechanisms play a major role in the execution dynamic and comprise activities like distribution of users, distribution of the computation effort, balancing parallel processes. In this respect, the framework can be extended with various algorithms for dynamic distribution and load balancing.

VIII. REFERENCES

[1] Md. Safat Hossain, “Performance evaluation web testing for E-commerce web sites”, IEEE/OSA/APR International Conference on Informatics, Electronics & Vision978-1-4673-1154-01.2012.

[2] Fares N. Almari, Pavol Zavarsky, Ron Ruhl, “Performance analysis of oracle database in virtual environments”, IEEE 26th International Conference on Advanced Information Networking and Applications Workshops. IEEE 978. -1-4503-1711.2012.

[3] Nuno Laranjeiro, Marco Vieira, “A robustness testing approach for SOAP web services”, The Brazilian Computer Society, J Internet Serve Apple (2012) 3:215–232 DOI 10.1007/s13174-012-0062-2012.

[4] Alexey Cheptsov, “OmpiJava - A Tool for development of high-performance reasoning applications for the semantic web”, ACM Web-KR’12, Maui, Hawaii, USA, ACM 978-1-4503-1711-5- October 29, 2012.

[5] Martti Vassar, Satish Narayana Srirama, “Framework for monitoring and testing web application scalability on the cloud”, ACM WICSA/ Helsinki, Finland, ACM 978-1-4503-1568-2012.

[6] Stavros Papastavrou, Panos K. Chrysanthis, “Performance vs. freshness in web database applications”, Springer, Wide Web 17:969–995 DOI 10.1007/s11280-013-0262-0 2014.

[7] Fredrik Abbors, Tanwir Ahmad, “Model-based performance testing in the cloud using the MBPeT tool”, ACM, ICPE’13, Prague, Czech Republic. ACM 978-1-4503-1636-1/13/04.2013.

[8] Michael Pradel, Parker Schuh, “Event Break: Analyzing the responsiveness of user interfaces through performance-guided test generation”, ACM-978-1-4503-2585DOI:10.1145/2660193.2660233 2014.

[9] Xiaolin Xu, Hai Jin, “URMG: Enhanced CBMG-Based Method for Automatically Testing Web Applications in the Cloud”, TSINGHUA SCIENCE AND TECHNOLOGY, ISSNl1007-0214l l07/10l lpp65-75 Volume 19, Number 1, February 2014.

[10] Luiz H. Nunes, Luis H. V. Nakamura, “PESOS: A Web Tool for Planning and Execution of Experiments in Service-Oriented Systems”, IEEE International Conference on Web Services, IEEE 978-1-4799-5054-6 DOI 10.1109/ICWS.2014.

[11] Joy deep Mukherjee, Mae Wang, “Performance Testing Web Applications on the Cloud”, and IEEE International Conference on Software Testing, Verification, and Validation Workshops, IEEE978-0-7695-5194-4/14DOI 10.1109 / ICSTW.2014.

[12] Amira Ali, Nagwa Badr, “Performance Testing as a Service for Web Applications”, IEEE Seventh International Conference on Intelligent Computing and Information Systems (ICICIS' I 5), 2015.

[13] Hidam Kumarjit Singh, “Performance Metrics of a Customized Web Application Developed for Monitoring Sensor Data”, IEEE 2nd International Conference on Recent Trends in Information Systems (Re TIS), IEEE978--4799-8349-0/15/ 2015.

[14] Ioannis K. Chaniotis, “Is Node.js a viable option for building modern web applications? A performance evaluation study”, Springer, Cross mark, Computing (2015) 97:1023–1044 DOI 10.1007/s00607-014-0394-9 2015.

[15] Javier Verd_u and Alex Pajuelo, “Performance Scalability Analysis of JavaScript Applications with Web Workers”, IEEE Computer architecture letters, VOL.15, NO.2, July-December 2016.

[16] Chia-Hung Kao, “Testing and evaluation framework for virtualization Technologies”, Springer, Cross mark computing, DOI10.1007/s006 07-016 - 0517-6 2016.

[17] Nuno Antunes, “Designing vulnerability testing tools for web services: approach, components, and tools”, Springer, Cross Mark, Int. J. Inf. Secure. DOI 10.1007/s10207-016-0334-0 2016.

[18] G.Deepa, P.Santhi Thilagam, “Black-box detection of X Query injection and parameter tampering vulnerabilities in web applications”, Springer, Cross Mark, Int. J. Inf. Secure. DOI 10.1007/s10207-016-0359-4 2017.

[19] B.A.Pozin and I. V. Galakhov, “Models in Performance Testing”, ISSN 0361_7688, Programming and Computer Software, 2011, Vol.37, No.1, pp. 15–25. © Pleiades Publishing, Ltd., 2011.

[20] Julio Cezar Estrella, Andre Takeshi Endo, Rubens Kenji T. Toyohara, Regina H. C. Santana, Marcos J. Santana, Sarita Mazzini Bruschi, “A Performance Evaluation Study for Web Services Attachments”, IEEE International Conference on Web Services, 978-0-7695-3709-2/09, DOI 10.1109/ICWS.2009.

[21] Jeff Offutt · Ye Wu, “Modeling presentation layers of web applications for testing”, Springer-Verlag, Soft Syst. Model (2010)9:257–280, DOI 10.10 07/s10270-009-0125-4, 2009.

[22] Md Umar Khan and Dr. T.V. Rao, “Web Application’s reliability improvement through architectural patterns”, International Journal of Web & Semantic Technology (IJWesT) Vol.5, No.3, DOI:10.5121 /ijwest 2014.53 01 July 2014.

[23] Guangzhou Jiang, Shujuan Jiang, “A Quick Testing Model of Web Performance Based on Testing Flow and its Application”, Sixth Web Information Systems and Applications Conference, IEEE 978-0-7695- 3874-7/09 DOI 10.1109/WISA.2009.16.

[24] Osama Hamed and Nedal Kafri, “Performance Testing for Web-Based Application Architectures (.NET vs. Java EE)”, IEEE 978-1-4244-4615-5/09/2009.

[25] Qinglin Wu, Yan Wang, “Performance Testing and Optimization of J2EE-based Web Applications”, Second International Workshop on Education Technology and Computer Science, 2010.

[26] J. Križani, A. Gregori, M. Mošmondor, P. Lazarevski, “Load testing and performance monitoring tools in use with AJAX-based web applications”, MIPRO 2010, May 24-28, 2010, Opatija, Croatia.

[27] Kunhua Zhu Junhui Fu Yancui Li, “Research the performance testing and performance improvement strategy in web application”, 2nd International Conference on Education Technology and Computer (ICETC), IEEE 978-1-4244-6370-11 2010.

[28] Martin Pinzger, Gabriele Kotsis, “AWPS- Simulation-Based Automated Web Performance Analysis and Prediction”, Seventh International Conference on the Quantitative Evaluation of Systems, IEEE 978-0-7695-4188-4/10 DOI 10.1109/QEST.2010.32.2010.

[29] M. Kalita, T. Bezboruah, “Investigation on performance testing and evaluation of PReWebD: a .NET technique for implementing web application”, The Institution of Engineering and Technology, IET Software. 2011, Vol. 5, Issue. 4, pp. 357–365 357 DOI: 10.1049/iet-sen.2010.0139. 2011.

[30] Andrea Adamoli, Dmitrijs Zaparanuks, Milan Jovic, Matthias Hauswirth, “Automated GUI performance testing”, Springer Science + Business Media, LLC, Software Qual J (2011) 19:801–839 DOI 10.1007/s11219-011-9135-x.2011.

[31] Leslie Cheung, Leana Golubchik, Fei Sha, “A Study of Web Services Performance Prediction: A Client’s Perspective”, 19th Annual IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, IEEE Computer Society 1526- 7539/11 DOI 10.1109/MASCOTS.66.2011.

[32] Nuno Laranjeiro · Marco Vieira · Henrique Madeira, “A robustness testing approach for SOAP Web services”, Springer Internet Serve Appl (2012) 3:215–232 DOI 10.1007/s13174-012-0062-2012.

[33] Madhvi Kamra, Ratnamala Manna, “Performance of Cloud-based Scalability and Load with an Automation Testing Tool In Virtual World”, IEEE Eighth World Congress on Services, IEEE Computer Society, IEEE 978-0-7695-4756-5/12 DOI 10.1109 /SERVICES .54.2012.

[34] Clyde Robson, Christian Bohm, Pawel Plucinski and Samuel Silverstein, “High-Performance Web Applications for Secure System Monitoring and Control”, IEEE Nuclear Science Symposium and Medical Imaging Conference Record (NSS/MIC), IEEE 978-1-4673-2030-6/12/2012.

[35] Stavros Papastavrou, Panos K. Chrysanthis, George Samaras, “Performance vs. freshness in web database applications”, Springer, World Wide Web (2014) 17:969–995 DOI 10.1007/s11280-013-0262-0.2014.

[36] G. Vadivelou, E. Ilavarasan, “Performance evaluation of semantic approaches for automatic clustering of similar web services”, World Congress on Computing and Communication Technologies, CPS, IEEE 978-1-4799-2876-7/13 DOI 10.1109/WCCCT.41.2014.

[37] Serdar Do˘gan, Aysu Betin-Can, Vahid Garousi, “Web application testing: A systematic literature review”, Do˘gan et al. / The Journal of Systems and Software 91-2014.

[38] Nitin Naik, Paul Jenkins, Philip Davies, and David Newell, “Native Web Communication Protocols and Their Effects on the Performance of Web Services and Systems”, IEEE International Conference on Computer and Information Technology, IEEE 978-1-5090-4314-9/16 DOI 10.1109/CIT .100.2016.

[39] M. Shaban Jokhio, Jing Sun, Gillian Dobbie, Tianming Hu, “Goal-based testing of semantic web services”, Elsevier DOI10.1016/j.infsof.2016 .11.011 0950-5849/2016.

[40] Igor Jugo, Dragutin Kermek, and Ana Me˘strovi´c, “Analysis and Evaluation of Web Application Performance Enhancement Techniques”, Springer International Publishing Switzerland 2014, ICWE 2014, LNCS 8541, pp. 40–56, 2014.

[41] Pushpendra Kumar Singh, Prabhakar Gupta, S.S. Bedi, and Krishna Singh, “Analyze the Performance of New Edge Web Application’s over N-Tiers Layer Architecture”, Springer-Verlag Berlin Heidelberg 2011, HPAGC 2011, CCIS 169, pp. 299–305, 2011.

[42] Neha Shankar Das, Mavra Usmani, Shivani Jain, “Implementation and Performance Evaluation of Sentiment Analysis Web Application in Cloud Computing Using IBM Blue mix”, International Conference on Computing, Communication and Automation (ICCCA2015), IEEE 978-1-4799-8890- 7/15/2015.

[43] Sandhya Kiran, Akshyansu Mohapatra, Rajashekara Swamy, “Experiences in Performance Testing of Web Applications with Unified Authentication Platform using JMeter”, International Symposium on Technology Management and Emerging Technologies (IST MET), IEEE 978-1-4799-723-5/15/ August 25 - 27, 2015.

[44] Connor Imes, David H.K. Kim, Martina Maggio, Henry Hoffmann, “Portable Multicore Resource Management for Applications with Performance Constraints”, IEEE 10th International Symposium on Embedded Multicore/Many-core Systems-on-Chip Sixth Web Information Systems and Applications Conference, IEEE 978-1-5090-3531-1/ DOI 10.1109/ MCSoC 2016.

[45] Robert B. Watson, Jan H. Spyridakis, “The Need For Innovation in Assessing Web Content Performance: A Case Study”, IEEE 978-1-5090-1761-4/16/2016.

[46] Elder M. Rodrigues, Maicon Bernardino, Leandro T. Costa, Avelino F. Zorzo, Flávio M. Oliveira, “PLeTsPerf - A Model-Based Performance Testing Tool”, IEEE 978-1-4799-7125-1/15/2015.

[47] B. Vani, R. Deepalakshmi, S. Suriya, “Web-based Testing – An Optimal Solution to Handle Peak Load”, IEEE 978-1-4673-5845-3/13/2013.

[48] Piero Fraternali and Massimo Tisi2, “Multi-level Tests for Model Driven Web Applications”, Springer-Verlag Berlin Heidelberg, ICWE 2010, LNCS 6189, pp. 158–172, 2010.

[49] Alessio Gambi, Giovanni Toffetti, and Sara Comai, “Model-Driven Web Engineering Performance Prediction with Layered Queue Networks”, Springer-Verlag Berlin Heidelberg, ICWE 2010 Workshops, LNCS 6385, pp. 25–36, 2010