A Survey of Most Common Referred Automated Performance Testing Tools

1 Muhammad Sadiq, 2 Muhammad Shahid Iqbal, 3 Amizah Malip, 4 Wan Ainun Mior Othman

1 Faculty of Computing (RIU) Islamabad
2 School of Computer Science, Anhui University, Hefei, China
3 Institute of Mathematical Science, University of Malaya

1 sadiq.paec@riu.edu.pk, 2 nawabishahid@yahoo.com, 3 amizha.malip@um.edu.my, 4 wanainun@um.edu.my

ABSTRACT

System performance is one of the key accounts in any software quality. Automated software performance can be measured with the help of testing based upon some predefine characteristics. Automated performance testing is very important for large scale and distributed applications. Unsatisfactory performance may create functional and non-functional problems which must lead to inference in terms of time and resources. In standards regulation authorities have propose matrix for automated software performance testing. In this discuss some software quality standards and their purposed matrix. We review some literatures for performance measurement based upon some characteristics or matrix. We then combined all characteristics in the form of matrix. We analyze some most commonly referred automated software performance testing tools to verify the practical implementation of purpose matrix.

Keywords: Automated, performance testing, response time, parameters

1. INTRODUCTION

Testing can be divided into two types which are functional and non-functional [11], [122]. Functional tests can determine whether an application achieves what it is expected to from a business perspective. It is conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements.

Functionality testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic.

Meanwhile performance elaborates non-functional [122].

Software product testing has great importance in an error detection of a software development it reflects directly on the software quality enhancement before its wrong implementation. Testing is widely used in industry for quality assurance [32]. Software testing is a formal process carried out by a specialized testing team. The team will examine software unit, several integrated units or an entire software package by running the programs on a computer. All the associated will be performed according to approved test procedures [76]. In software development life cycle, software testing is highly recommended to assure the quality of the software process and product. ‘Researchers and practitioners’ came-up with a variety of different software tools to automate the testing process.

Literature survey shows that two appropriate methods are commonly used to measure for performance testing. Which are Load testing and stress testing. Load testing assesses how a system performs under a given “load.” The rate at which transactions are submitted to the system is called the load. One of load testing objectives is to determine the maximum sustainable load the system can handle [121]. Meanwhile web application load is defined in terms of concurrent users or HTTP connections. “Stress testing refers to subject a system to an unreasonable load with the intention of breaking it. A stress test denies a system the resources (e.g., RAM, disk, interrupts, etc.) needed to process a certain load. It is designed to cause a failure. It tests the system’s fault recovery capability [121]. Trying to break the system under test by overwhelming its resources or by taking resources away from it. The purpose is to make sure that the system fails and recovers gracefully as double the baseline number of concurrent Users/HTTP connections.

Randomly shut down and restart ports on the network. If you cannot measure it, you cannot improve it [26]. We analyze our literature with this concept. We will find commonly used approaches to measure the performance. ‘Section two describe, the existing methodologies or standards and they proposed matrix, meanwhile section three discuss literature review. On section four presented most common refers tools, their features and implementation description. In section five we describe proposed metrics and guidelines for the development of automated performance testing tools.

2. EXISTING STATE OF THE ART/METHODOLOGY

There are number of quality monitoring organization and agencies which monitor guides and regular standards to measure the parameters. In this regard we check the available standards of software performance which is IEEE STD 1061 -1992. In 1998 another revision some revision were make but parameters for performance remains same. Another revision was 2009 however the parameters for performance remain as per previous IEEE Standard for a Software Quality Metrics Methodology describes the performance base on these parameters.
As per ISO/IEC 9126 performance is measure in terms of efficiency meanwhile the parameters for measure are time behavior, resource utilization and efficiency compliance. In 2010 ISO/IEC FDIS 25010:2010(E) slandered efficiency changes to performance efficiency and the parameters are time behavior, resource utilization and capacity. Time behaviors refer to response time and throughput.

Performance efficiency as per ISO/IEC FDIS 25010:2010(E).

Performance relative to the amount of resources used under stated conditions.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Efficiency</td>
<td>Relative extent to which a resource is utilized (for example, storage space, processing time, or communication time).</td>
</tr>
<tr>
<td>Integrity</td>
<td>Extent to which the software will perform without failures due to unauthorized access to the code or data within a specified time period.</td>
</tr>
<tr>
<td>Reliability</td>
<td>Extent to which the software will perform without failures within a specified time period.</td>
</tr>
<tr>
<td>Survivability</td>
<td>Extent to which the software will perform and support critical functions without failures within a specified time period when a portion of the system is inoperable.</td>
</tr>
<tr>
<td>Usability</td>
<td>Relative effort for training or software operation (for example, familiarization, input preparation, execution, or output interpretation).</td>
</tr>
</tbody>
</table>

3. RELATED WORK

A large amount of research has been done over the last few decades to address the problem of analyzing the performance of software systems. In general, much of the work describes methodologies that include modeling the software systems, capturing the performance characteristics and transforming the software specification models into performance models. Gathering data to assess the performance is an essential aspect of software performance engineering approach. In this section, we review some of the methodologies and tools used for software performance assessment and techniques used for gathering and capturing performance parameters (data), which are highly dynamic and uncertain. Flippines, I. vokolos et al.[1998] uses response time, throughput, latency and resource utilization for software performance testing [22] and in [2000] introduces a new approach with same parameters [20]. Huebner [2001] discussed performance testing for IP services and systems via load testing and stress testing based on response time and resource utilization [121]. Lundberg et al. [2001] discussed the conflicts and Trade-Offs between software performance and maintainability and the focus performance matrix were response time, throughput, and resource utilization and compare the real time system and normal routine systems. Ulrich Herzog (2001) present a new technique of performance validation for software/hardware systems based on response time, throughput, latency and reliability [12]. Giovanni Denaro and Andrea Polini (2004) applies performance testing on distributed applications based on response time, throughput, scalability and latency parameters [23].

Nenad Stankovic (2006) discussed some patterns and tools for performance testing the tool address functional aspect of test case implementation and execution and behavioral and automation issues [3].

Vibhu Saujanya Sharma (2006) quantify software performance base on response time, latency, throughput resource utilization, scalability, reliability and security and discuss the trade Offs between various attributes and the importance of optimize system configuration [1].


Gwang-hun Kim and Hui-choun Moon (2009) discussed the software performance scheme using virtualization technology based on time behavior, scalability and resource utilization [86]. Hangjuang Zo (2010) discussed the security and performance in service oriented applications based on response time, latency, throughput and security and also discussed the tradeoff among these parameters [14]. Li Yuan yu (2010) presents a method to test Linux software performance the focus parameters was time behavior, scalability, and resource utilization [18]. H. Sarojadevi (2011) discussed
the methodologies and tools for software performance testing the focus parameters were response time, throughput, availability, scalability and also discussed some familiar tools [127].


4. MOST COMMON REFERRED PERFORMANCE TESTING TOOLS

4.1 Grinder: [55], [33], [56], [38], [49], [58], [59], [60].

The Grinder works on any hardware platform and any operating system that supports J2SE 1.4 and above. It can simulate web browsers and other devices that use HTTP, HTTPS, and it can be used to test Web Service interfaces using protocols such as SOAP and XML-RPC, databases using JDBC, MOM based systems using different types of protocols such as IIOP, RMI/IIOP, RMI/JRMP, and JMS. It can be used to test systems that utilize other protocols such as POP3, SMTP, FTP, and LDAP [94].

4.1.1 Response Time, Throughput, Latency

On the Grinder’s console there are two tabs which display information about The Grinder and its tests.

1st are graphs and 2nd are results. Grinder analyzer parses the data from grinder and generates Graph for response time, throughput, and latency. When user perform the test grinder shows Statistics, Reports, Charts to represent the response time, throughput and latency for tests [96], [97], [95], [98].

4.1.2 Scalability

After performing test the grinder shows the bandwidth used by different test and different scenarios at different load level like response time at different loads etc. [96].

4.1.3 Resource Utilization

Grinder generates different Graphs at different loads which shows the bandwidth usage and resource utilization like memory, bandwidth and CPU utilization [96], [97], [95], [98].

4.1.4 Security

Grinder provides HTTP/J2EE form based authentication which provides a more complex HTTP example based on an authentication conversation with the server. This HTTP script describes the path based response and J2EE Servlets describes common model for form based authentication. When user tries to access a secured resource J2EE servlets challenges with logon page and logon page POSTs to J security check page. [96], [97], [95], [98].

4.2 Apache JMeter: [43], [40], [51], [67], [68], [55], [73] [69], [70], [38], [57], [71], [72], [60], [67].

Apache JMeter is a powerful desktop performance testing tool from the Apache Jakarta project, written in Java, for load-testing web pages, web applications, and other static and dynamic resources including databases, files, Servlets, Perl scripts, Java Objects, FTP Servers, and more. The main component in JMeter is the ‘Java Swing-based Graphical User Interface’ which can be used for both Scripting and Execution [99], [100], [101], [102], [103], [104].

4.2.1 Response Time

Apache JMeter provides the response time of tests at different load point and also provides the average response time of each test [99], [100], [101],[102], [103], [104].

4.2.2 Throughput

JMeter have an option of summary reports which creates a table row for each request. The throughput can be calculated from the point of view of the sampler target (e.g. the remote server in the case of HTTP samples).

JMeter stores the total time which the request have taken in his account. So if other samplers and timers are belonging from the same thread, this will increase automatically the total time, so it will reduce the throughput value. With this scenario we can say that two identical samplers with different names will have half the throughput of two samplers with the same name. So it is very important to choose the sampler labels correctly which will help to get the best results from the Report [99], [100], [101], [102], [103], [104].

4.2.3 Latency

The summary report of apache JMeter show the latency of each request at different loads level [99], [100], [101], [102], [103], [104].

4.2.4 Scalability

Apache JMeter is used for Load testing and stress testing. With the help of these tests we can measure the scalability of our product because we can test our
product with multiple users at a time. [99], [100], [101], [102], [103], [104].

4.2.5 Resource Utilization

Report analyzer tells us about the CPU and other resource usage at different load level [99], [100], [101], [102], [103], [104].

4.3 Silk performer: [85], [52], [86], [57], [87], [58], [88].

4.3.1 Response Time

In reports The Transactions section contains summary measurements in a tabular form, that is, aggregate measurements for all transactions of the specific user. For every transaction, the transaction response time and the transaction busy time are displayed. The transaction response time is measured from the beginning to the end of the transaction. [105], [106], [107], [108].

4.3.2 Throughput

Summary table for each user group provides detailed measurements in tabular form. The measurements include transaction response times, individual timers, counters, and response time and throughput measurements related to the type of application that was tested (Web, database, CORBA, or TUXEDO). In addition, errors and warnings for all user groups [105], [106], [107], [108].

4.3.3 Scalability

Silk performer Ensure the scalability, performance, and reliability of your enterprise applications. Silk Performer ensures the quality of your enterprise applications by measuring their performance from the end-user perspective, as well as internally, in a variety of workload scenarios and dynamic load conditions [105], [106], [107], [108].

4.3.4 Resource Utilization

Log file for each test is created which shows the report regarding the tests including resource utilization [105], [106], [107], [108].

4.4 Mercury Interactive Load Runner: [42], [78], [44] [46], [48], [49], [39], [41], [65].

4.4.1 Response Time

The Transaction monitor displays the transaction rate and response time during scenario or session step execution. Transaction Response Time - Whole Scenario graph, lets you monitor the Amount of time it takes for each transaction to be completed [109], [110], [111].

4.4.2 Throughput

The Throughput graph shows the amount of data (measured in bytes) that the Users receive from the server at any given second. You can compare this graph with the Transaction Response Time graph to see how throughput affects transaction performance [109], [110], [111].

4.4.3 Latency

FTP Pass Latency shows the Interval between transmitting a FTP PASS packet and receiving a response in msec [109], [110], [111].

4.4.4 Scalability

Running Users - Whole Scenario graph displays the number of Users running and the behavior of Users at a given time [109], [110], [111].

4.4.5 Resource Utilization

Windows Resources graph displays the Windows resources measured during a scenario [109], [110], [111].

4.4.6 Security

When you run certain security scripts, you can use Load runner’s security graphs to view information about the simulated attacks on the server [109], [110], [111].

4.5 IBM Rational Performance Tester: [61][62][63][65][66]

4.5.1 Response Time

Select the Page Performance tab. This tab presents a bar graph of the average page response time for the 10 pages with the highest times. The Summary page displays the following Page Summary information. The average response time for all pages. The maximum response time for all pages, the minimum response time for all pages [112], [113], [114], [115].

4.5.2 Throughput

Byte counters provide throughput information regarding the rate and the number of bytes sent and received during a sample interval and during a run [112], [113], [114], [115].

4.5.3 Resource Utilization

There are reports to measure network and CPU utilization, to identify the presence of hardware bottlenecks. You can monitor your resources via reports [112], [113], [114], [115].

4.5.4 Scalability

During load testing the no of users and behavior of application at different loads enable to scale the product for optimize use [112], [113], [114], [115].

4.5.5 Security

In IBM Rational Performance tester user can define his security algorithms by implementing custom security Java™ interfaces which can be used in the security editor. With customize security algorithms. User can implement proprietary security algorithms that transform the XML before sending to and after receiving from the server [112], [113], [114], [115].
4.6 Open STA. [123][124][125][74][75][77]

4.6.1 Response Time
Open STA provides different types of data collection and monitoring functions. When user runs a Test it generates wide range of results data which is collected automatically tool. It includes response times of virtual users and resource utilization of r test. User can also create and reference Collectors in for his Tests to enhance the Test-run monitoring. Data collection options are also available [117], [118], [119], [120].

4.6.2 Throughput
Its NT Performance Collectors collects the performance data during the Execution of test from performance objects. Each performance object has set of performance counters which are associated with performance objects. It helps to measure throughput [117], [118], [119], [120].

4.6.3 Latency
Min Request Latency indicates that the minimum length of time elapsed in milliseconds between sending of an HTTP request and receiving the results of request [117], [118], [119], [120].

4.6.4 Scalability
The behavior of application at different loads helps us to measure the scalability of application. The resultant graphs show the total and average behavior and bandwidth used [117], [118], [119], [120].

4.6.5 Security
A proxy server acts as a security barrier between your internal networks (Intranet) and the Internet, keeping unauthorized external users from gaining access to confidential information on your internal network. This is a function that is often combined with a firewall [117], [118], [119], [120].

4.7 HP Load Runner: [44], [52], [55], [39], [73], [74], [84], [75], [65], [60], [76],[77]
Load Runner is a software testing tool which enables you to test a range of applications including mobile, Ajax, Flex, HTML 5, .NET, Java, GWT, Silverlight, SOAP, Citrix, ERP and legacy. It isolated performance bottlenecks with system and end-user monitoring and a wide range of analysis components.[91]

4.7.1 Response Time
A view of HP Load Runner Analysis cross-results trending capabilities, showing a comparison of increased System scalability and optimized response time performance. We can monitor the response time of requests with the help of resultant graph. Transaction Response Time Whole Scenario graphs. Shows the amount of time it takes for each transaction to be completed. [92], [93]

4.7.2 Throughput
The Throughput graph shows the amount of data (measured in bytes) that the users receive from the server at any given second. You can compare this graph with the Transaction Response Time graph to see how throughput affects transaction performance. If the throughput scales upward as time progresses and the number of users increase, this indicates that the bandwidth is sufficient. If the graph were to remain relatively flat as the number of users increased, it would be reasonable to conclude that the bandwidth is constraining the volume of data delivered [93].

4.7.3 Latency
Hits per Second - Whole Scenario graph. Displays the number of hits (HTTP requests) made to the Web server by users during each second of the scenario run. And also shows the delays [93].

4.7.4 Scalability
Running users - Whole Scenario graph displays the number of users running at a given time [93].

4.7.5 Resource Utilization
Windows Resources graph displays the Windows resources measured during a scenario [93].

5. PURPOSED STATE OF THE ART/METHODOLOGY
After literature review we find some common points to establish slandered criteria for performance measure. To measure the software performance of any kind of software all these parameters are necessary but the level of assurance depends on user requirements.

5.1 Response Time
Response time which is defined as the duration from when a client makes a request until the entire file is received by the client [8].

5.2 Throughput
Throughput denotes the number of operation that can be completed in a given period of time [23].

5.3 Latency
Latency Typically Describes the Delay between request and completion of an event [23].

5.4 Scalability
Scalability is the ability of a system to continue to meet its response time or Throughput objectives as the demand for the software function increases”[27] “Scalability identifies the dependency between the no of distributed system resources that can be used by an application (typically no of hosts) [23].

5.5 Resources Utilization
Typical Resources that considered include network bandwidth requirement, CPU cycle, disk space, disk access operation, and memory usage [10], [11].
5.6 Security

Ensure that the security requirements are meeting [7].

Table 1: Common Referred tools and feature mapping

<table>
<thead>
<tr>
<th>S.No</th>
<th>Name of Tools</th>
<th>Parameters</th>
<th>Response Time</th>
<th>Latency</th>
<th>Throughput</th>
<th>Scalability</th>
<th>Resource Utilization</th>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>HP Load runner</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>P</td>
<td></td>
<td>✓</td>
<td>X</td>
</tr>
<tr>
<td>2</td>
<td>Grinder</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
<td>P</td>
<td>X</td>
</tr>
<tr>
<td>3</td>
<td>Apache JMeter</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>P</td>
<td>X</td>
</tr>
<tr>
<td>4</td>
<td>Silk performer</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>P</td>
<td>P</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>5</td>
<td>Mercury Interactive Load runner</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
<td>P</td>
<td>X</td>
</tr>
<tr>
<td>6</td>
<td>IBM rational Performance</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>P</td>
<td>✓</td>
<td>P</td>
<td>X</td>
</tr>
<tr>
<td>7</td>
<td>Open STA</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

In this table 1 we can easily identify tools that are not implementing parameters of performance testing. Some tools archive partial implementation such as web application. While some are implementing a single aspect of parameter and some are totally neglecting the standard defined parameters. Some tools are implementing some extra features so it is recommended to add such type of features in standards due to its wide utilization in industries. It can easily noticed that response time, latency and throughput is implemented by all most common referred tools but scalability, resource implementation and security is not implemented by all common referred tools.

Security is very key component of performance testing with respect to current era.

Figure 1 shows the detail overview of tools with respect to features implementations. We can say figure 1 is graphical representation of table 1. Mercury Interactive load runner is comparative better because it is implementing more performance parameters then other tools.

Figure 1

The Figure shows tools and implementation of parameters. HP Load runner, Grinder and Apache JMeter implement four parameters completely and one partially. Silk performer and Open STA are least implement parameters. IBM rational Performance implements four parameters and two partially. Mercury Interactive Load runner implements complete five parameters.
6. CONCLUSION

There are so many tools are available but not a single which are completely implement the parameters of performance.

In this paper we discuss tools base performance process and their parameters. In this paper no signal tool have completely implement the whole parameters. Also in this paper we identifies the parameters which are implemented any automated performance testing tools.

This paper helps for the development of performance testing tools. Analysis some most common referred automated software performance testing tools from literature to verify the practical implementation of purposed matrix. In this way we can identify gaps of other testing tools like static testing, functional testing, test case management etc. In this way we can identify the gap and provide guidelines to develop tools in future in better way.

REFERENCES


[41] Osama Hamed1 and Nedal Kafri, "Performance Testing for Web Based Application Architectures (.NET vs. Java EE)" ©2009 IEEE.


[82] Gordon Fraser, Andrea Arcuri;” Whole Test Suite Generation “IEEE TRANSACTIONS ON SOFTWARE ENGINEERING 2012.


[90] Jimoh, R.g & Abikoye, O.C. ;”SOFTWARE PERFORMANCE EVALUATION ALGORITHM EXPERIMENT FOR IN-HOUSE SOFTWARE USING INTER-FAILURE DATA” International journal of engineering and management sciences 2012


[94] http://grinder.sourceforge.net/g3/features.html#Sta ndards last access = 16-12-2012

[95] http://grinder.sourceforge.net/g3/tutorial-perks.html last access = 16-12-2012


Tiantian Gao; Yujia Ge; Gongxin Wu; Jinlong Ni


Amanpreet Kaur “Outsourcing Software quality” Norwegian University of Science and Technology Department of Computer and Information Science, 2013


Amanpreet Kaur “Outsourcing Software quality” Norwegian University of Science and Technology Department of Computer and Information Science, 2013

