

Performance Analysis Using JMeter 5

¹Kusum, ²Maneesha

^{1,2}Department of Computer Science and Engineering, NC College of Engineering, Panipat, Haryana, India. *Corresponding Author: sditm12110@gmail.com

Abstract— Performance analysis methodology consumes multiple aspects like load testing, stress testing, volume testing etc. to analyse any software's performance and helps in delivering a fast and reliable software system. Performance analysis is used to improve user experience to any web, mobile or desktop application. Metrics of performance analysis of a software system are described and represented in terms of numerical values. Here we will understand to measure the performance statistics using JMeter and analyse various performance metrics. Software performance analysis using these performance metrics (like Availability, Mean Time to Failure, Mean Time Between Failure etc.) is very common in small, mid and large company software processing applications to understand the efficiency of the product in terms of its failure and availability.

Keywords— User Experience (UX), Performance Metrics, JMeter, User Interface (UI), Software Reliability

I. INTRODUCTION

Performance analysis is underpinned by systematic observation, which provides valid, reliable and detailed information related to performance of any system [1]. Performance can be easily represented in terms of performance metrics [2]. Basically, there are few major points that are kept in mind before the release of a product in the market which can include: functionality of the software as per the client requirements, software system security, performance of that software system etc. And, performance of a software system is very important to put a great impact on UX (User experience) and improve the efficiency of the software product [3].

Although there has been little work published that uses software reliability (UX) to measure software performance analysis, it is nonetheless an extremely significant issue for small to large industrial projects as the performance can spoil the user experience [4]. Often, the problems that any project faces after release are system crashes or incorrect system responses, and performance degradation can reduce required system throughput. When queried, it is not uncommon to learn that although the software system has gone through extensive functionality testing, it was never really tested to assess its expected performance.

The issues that have to be addressed when doing performance analysis, differ in a number of ways from the issues that must be addressed when doing functional testing [5]. These methodologies are drawn from experiences in testing the performance of different industrial software systems. Performance analysis is very useful for the following:

1. The design of test case selection or generation algorithms specifically intended to test for

performance criteria rather than functional correctness criteria.

2. The definition of metrics to assess the comprehensiveness of a performance test case selection algorithm for a given complex program [6].
3. The definition of metrics to compare the effectiveness of different performance testing strategies for a given program.
4. The definition of relations to compare the relative effectiveness of different performance testing strategies in general. This requires that we are able to say in some concrete way what it means for this performance testing strategy to be better than that one.
5. The comparison of different hardware platforms or architectures for a given software application.

Performance analysis methodologies of a software systems does not only reduce the efforts for increasing software system quality, it also increases the effectiveness of software system [7].

II. PERFORMANCE ANALYSIS ARCHITECTURE

Performance analysis metrics (methodologies to measure performance of a software system) are very important for any industry level project. As a software is generally get fame among clients because of its user interactive, effectiveness and quickness. In this busy era of society, persons will not think to spend time on any software application which is slow responsive and consuming large amount of data which absolutely consume extra time to load [8]. The core part of this performance analysis consists following:

- JMeter
- A webpage request (GET / POST)
- Mathematical tools knowledge

Here, mathematical tools knowledge may consider basic mathematics probability, mean, median etc. Generally, performance analysis is carried out in some simple steps which are shown in the diagram. Initially we give any website URL or any web page as an input to the JMeter tool. As we know that with the help of JMeter we can find out the response of that given web page URL under some specific conditions or environment (number of requests, time per request, total time for all the requests etc.) [9]. Generally, we use API testing (Application Programming Interface testing consumes a request header and check corresponding output of that request directly without performing any action like click on any button or insert data in text fields etc. to the website) in JMeter to accumulate the performance statistics of any particular request or response and put any assertion to verify the response time, load time, latency etc. [10]. Also, JMeter provide various graphical output representation ways like performance output in tabular form, graphical form etc. Now, this output of the JMeter (Performance statistics) work as an input for the Performance Metrics (Analysis) [11]. Various methodologies can be applied on these statistics to find the reliability of the software application.

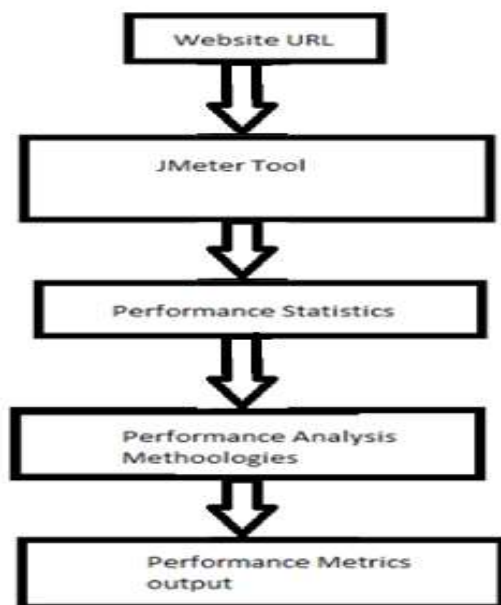


Figure 1: Performance Analysis Process

The above diagram depicts about the simple performance analysis steps. We will use API (Application Programming Interface) testing to analyse a single particular request and to measure a couple of requests, we will make a club of those APIs to create a session of requests. Here, performance metrics may be Availability of the software application, Mean Time to Failure (MTTF), Mean Time Between Failures (MTBF), Mean Time to Repair (MTTR)

etc. which will represent the quality of the application software.

This architecture will fulfil our objective to measure performance of any software application. Also, it will give us the quality of target software in terms of probability of percentage. This above architecture is the baseline of this research which is capable enough to track the performance of small size request to complex request and ensure user experience of target application on the basis of software reliability.

III. PROPOSED WORK

JMeter is an open source Java based software application which is used to test performance, functionality and load test of any web application [12]. Apache JMeter can be used for both static and dynamic websites. We can write scripting in JMeter to automate the performance testing, this adds one more advantage of using this software tool. JMeter is used for multipurpose like to display output reports in dynamic HTML format, to process JSON, HTML, XML, any plain text response, multithreading of requests etc. Apache JMeter 5.0 has improved user experience and bug fix over previous versions. One of the major advantages for many users is improvement in HTTP request. Earlier HTTP request can be reset to SSL state (SSL connections use certificate only connection creation time) but now HTTP request will reset at both SSL state and connection on each thread group iteration. We can process any file as multipart/form-data, JSON parameters can as well as simple parameters as request in JMeter 5.0. Also, we got some improvement over UI for user like output representation in the form of view tree and result status action handlers.

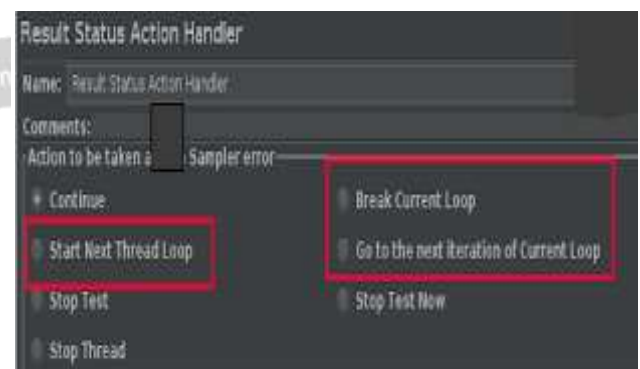


Figure 2: Result Status Action Handlers

Let us consider performance measuring and analysis of a website "https://www.tradegyani.com/". As, this URL is a get request and it will work as an input to the JMeter 5.0. Following steps would be carried out to analyse the performance of this web page:

Create a ThreadGroup: Initially, create new Thread

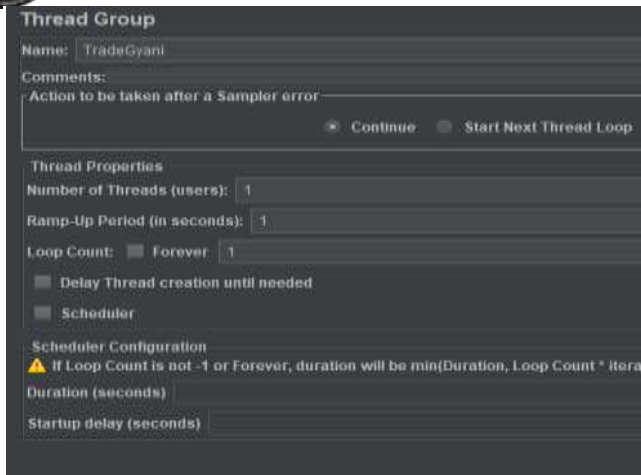


Figure 3: Thread Group

Group (name it as TradeGyani) under test plan in JMeter 5.0. We can also provide the total number of requests by passing number of threads(users).

Create HTTP request: Now, create an HTTP request under this Thread Group and give a suitable name to this request (like “MainRequest” as shown in following figure 4). Add the protocol for this web connection (like “https” in this scenario). Set server name or domain name of the webpage (like “www.tradegyani.com”). We can also give the relative path for the web page.

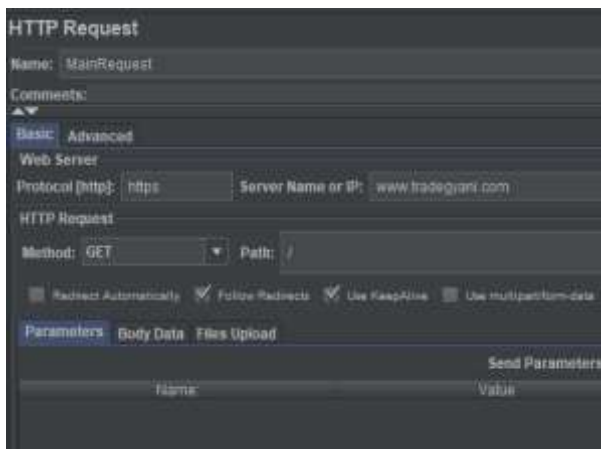


Figure 4: HTTP Request

Later on, set request method like GET, POST, HEAD etc. and set the relative path of the web page. As, we are using a get request so select “GET” and relative path as “/”. We can also provide request parameter, body data as well as file in same HTTP request as per requirement (leave all these empty in this example).

Add Listener: Add any listener to listen the process of JMeter and show output to the user. Here for this example let us add “View Results Tree” listener to capture the output of the web request. We can also store the output in system directory by passing the path to

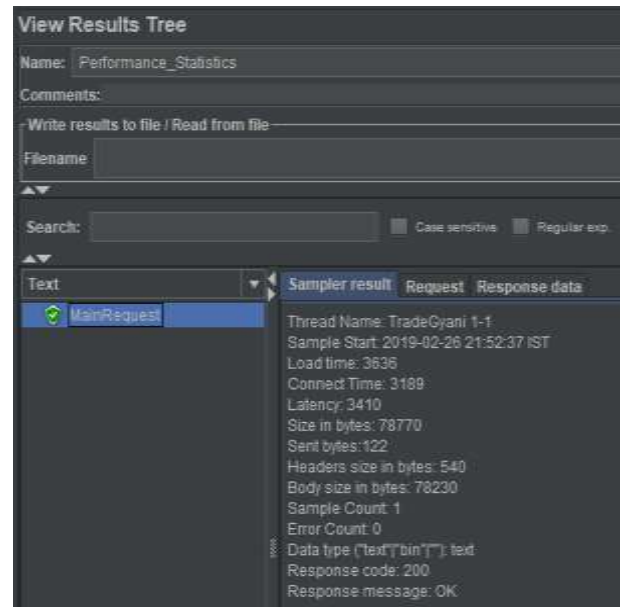


Figure 5: View Results Tree Listener

listener. We can also add multiple listeners for a single request.

Here, in JMeter response we can observe various time components as shown in above screenshot.

Load time is the difference of time when the actual request was sent from client and time when full response received. Load time is also known as sample time, elapsed time and response time [13]. Connect time is the time taken to establish TCP connection between client and server using TCP handshake. Latency can be defined as the time between the request sent from client and the first response received from server.

Save and Run JMeter script: Now, we just need to save the JMeter script (with “.jmx” extension) and run this script.

Capture performance statistics: Store these all response outputs like load time, connect time, Latency, request size etc. in any format (tabular/graphical/Pie chart etc.).

Analyse Performance Metrics: Now, calculate all the performance metrics like MTTR (Mean Time to Response), MTTF (Mean Time to Failure), MTBF (Mean Time Between Failure), Availability etc. We can use probability concept to evaluate these performance metrics to represent the performance of any web page or multiple web pages simultaneously. Use following formulae to measure performance metrics:

MTTF stands for the mean time duration between the length of the time for which the system was successfully running before any failure. MTTR is the mean time for amount of time to repair a system after failure or mean time to bring the system in working situation. And, MTBF can be calculated as:

$$MTBF = MTTF + MTTR$$

Performance Results: We can evaluate web application response time (captured by JMeter) as following:

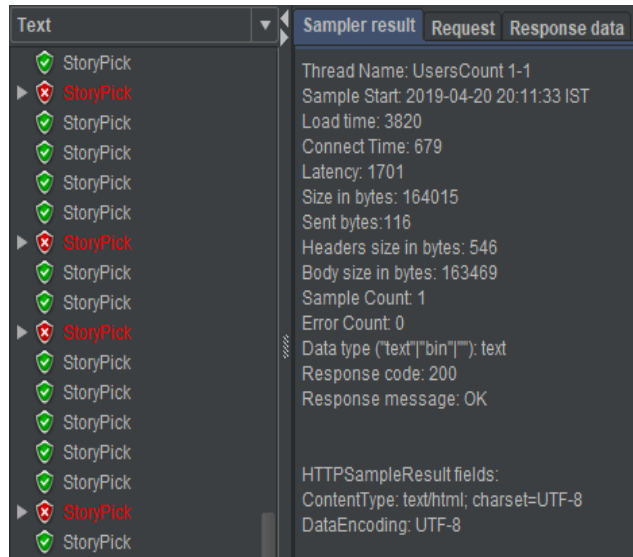


Figure 6: Sample Response time

IV RESULTS AND DISCUSSION

Following is the JMeter output in tabular form along with the manual readings captured in milli seconds (ms) for target web application:

Table 1: Performance Evaluation

Sr.No.	Manually recorded Response Time(in ms)	Jmeter Time Readings(in ms)				Cumulative Time Value
		Load Time	Connect Time	Latency	Response Time	
1	2550	8152	5833	7218	2319	8152
2	2200	2742	826	2095	1916	10894
3	1270	3456	956	2318	2500	14350
4	2220	2800	859	1256	1941	17150
5	2300	3084	774	2137	2310	20234
6	2060	3059	807	1927	2252	23293
7	4025	2307	868	0	1439	25600
8	2780	3773	733	1757	3040	29373
9	2240	3110	799	2081	2311	32483
10	2820	2326	818	0	1508	34809
11	2350	3853	991	2191	2862	38662
12	2360	3373	895	2176	2478	42035
13	2430	3446	879	2322	2567	45481
14	1283	3242	880	1880	2362	48723
15	2400	3272	905	1930	2367	51995
16	1544	2575	841	0	1734	54570
17	1767	3289	836	2187	2453	57859
Average	2270.529412	3403.471	1147.058824	1969.1	2256.411765	

Evaluation of Software Metrics:

Total number of requests = 17

Request failure count = 4

Request success count = 13

Let's assume that four failures are F1, F2, F3 and F4

Then, time to failures (in ms) occurrence is: 8153, 23294, 32484, 51996

Now, we can calculate Mean Time To Failure (MTTF) as following:

$$\text{MTTF} = (8153 + 23294 + 32484 + 51996) / 4$$

$$= 28981.75 \text{ ms}$$

Also, we get time to repair of the system for four failures is (in ms): 2742, 2307, 2326, 2575

So, we can calculate Mean Time To Repair (MTTR) as following:

$$\text{MTTR} = (2742 + 2307 + 2326 + 2575) / 4$$

$$= 2487.5 \text{ ms}$$

Now, calculating Mean Time Between Failures (MTBF) by using following:

$$\text{MTBF} = \text{MTTF} + \text{MTTR}$$

$$= 28981.75 + 2487.5$$

$$= 31469.25 \text{ ms}$$

$$\text{Availability} = \text{MTTF} / \text{MTBF}$$

$$= 28981.75 / 31469.25$$

$$= 0.92$$

Availability represents that here are the 92% chances for system availability of web application. This can ensure User experience along with user interface. As we all know that, in today's era organisations are focussing on User Interface / User Experience (UI / UX) instead of only user interface. And, performance metrics like MTTF, MTTR, MTBF, Availability ensures user experience to software application.

We can also evaluate various types of software testing to ensure more system performance concepts on the basis of these analysed performance metrics using JMeter 5.0 which are given below [14].

Scalability testing: The objective of scalability testing is to determine the software application's effectiveness in "scaling up" to support an increase in user load. It helps plan capacity addition to your software system.

Load testing: It checks the application's ability to perform under anticipated user loads. The objective is to identify performance bottlenecks before the software application goes live.

Stress testing: It involves testing an application under extreme workloads to see how it handles high traffic or data processing. The objective is to identify the breaking point of an application.

Volume testing: Under Volume Testing large no. of. Data is populated in a database and the overall software system's behaviour is monitored. The objective is to check software application's performance under varying database volumes.

Spike testing: It tests the software's reaction to sudden large spikes in the load generated by users.

Comparison with Earlier research: This research paper focussed on improvement of two aspects which are following:

- More accurate
- Better report generation / representation
- Better performance management [15]

We can differentiate earlier research with this research as per following table:

Table 2: Research work Comparison

Earlier Research	Proposed work
Jmeter is uable to report results.	Jmeter is smart enough to represent output in graphical, tabular, tree structure etc.
Jmeter accuracy was around 85% while working with more than one threads.	JMeter accuracy is around 90% because response time also include certificate installation time to ensure real time behavior.

V. CONCLUSION

Performance analysis and its management has become compulsory before the release of a software product to any particular business client or for open market. Performance analysis makes sure about the customer reliability and protects any investor's investment against the software product failure. Costs of performance analysis are usually more but it leads to improved customer satisfaction, loyalty, and retention to the organisation's software. Now a days, performance analysis has become a trend and an important part of the software testing because most of the companies are moving towards UI/UX (User Interface / User Experience) approach instead of only better UI approach.

Performance analysis does not improve only UX but it increases the overall quality of the software product and JMeter 5 helps us to measure those performance metrics within an optimum budget with more than 90% accuracy. Also, JMeter can represent the live performance progress of any website in a very simple graphical format without any manual effort in that case if we run JMeter in a loop. And, software reliability ensures the software product acceptance at user end in which JMeter plays an important role to evaluate that accepted UX term.

REFERENCES

- [1] Charmy Patel, Ravi Gulati, "Software Performance Analysis: A Data Mining Approach", IAEME, August 2014.
- [2] Tu Honglei, Sun Wei, Zhang Yanan, "The Research on Software Metrics and Software Complexity Metrics", International Forum on Computer Science-Technology and Applications, 2009.
- [3] Mohammad Zarour and Mubarak Alharbi, "User Experience Aspects and Dimensions: Systematic Literature Review", International Journal of Knowledge Engineering, Vol. 3, No. 2, December 2017.
- [4] Iqra Tariq, Talal Bin Maqsood, Baber Hayat, Khizar Hameed, Mobashirah Nasir, Maham Jahangir, "The comprehensive study on software reliability", International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), 2018.
- [5] Goshu, Y.Y. and Daniel Kitaw, D. (2017) 'Performance measurement and its recent challenge: a literature review', International Journal of Business Performance Management, May, 2016.
- [6] David Maplesden, Ewan Tempero, John Hosking, John C. Grundy, "Performance Analysis for Object-Oriented Software: A Systematic Mapping", Volume: 41 Issue: 7, January 2015.
- [7] Isha Arora, Vikram Bali, "A Brief Survey on Web Application Performance Testing Tools Literature Review", International Journal of Latest Trends in Engineering and Technology, May, 2015.
- [8] Pooja Ahlawat and Sanjay Tyagi, "A Comparative Analysis of Load Testing Tools Using Optimal Response Rate", Volume 3, Issue 5, May 2013.
- [9] Eljona Proko and Ilia Ninka, "Analyzing and Testing Web Application Performance", Dept. of Computer Science, Vol.3, Issue 10, October 2013.
- [10] Sukhdev Singh Ghuman, "Software Testing Techniques", International Journal of Computer Science and Mobile Computing, Vol. 3, Issue. 10, October 2014.
- [11] Reenu Bhatia, Anita Ganpati, "Performance Evaluation of JMeter, LoadComplete and WAPT", International Journal of Scientific & Engineering Research, Volume 7, Issue 12, December 2016.
- [12] Shikha Dhiman, Pratibha Sharma, "Performance Testing: A Comparative Study and Analysis of Web Service Testing Tools", IJCSMC, Vol. 5, Issue. 6, pg.507 – 512, June 2016.
- [13] Avritzer and E.J. Weyuker, "Deriving Workloads for Performance Testing" Software-Practice and Experience, vol. 26, no. 6, pp. 613-633, June 1996.
- [14] Muhammad Abid Jamil, Muhammad Arif, Normi Sham Awang Abubakar, Akhlaq Ahmad, "Software Testing Techniques: A Literature Review", 6th International Conference on Information and Communication Technology for The Muslim World, 2016.
- [15] Sandeep Gudla, Valli Sri Krishna Veni, IOSR Journal of Business and Management, Volume 6, Issue 3, Dec. 2012.