STRESS TESTING OF WEBSITES USING JMETER TESTING TOOL

[1]Dr. Y R Ghodasara, [2]Dr. K C Kamani , [3]Dr. G J Kamani, [4]Mr. P S Parsania

[1]Professor, [2]Asst. Professor, [3]Asst. Professor, [4]Asst. Professor
[1]Department of Agricultural Information Technology,
[1]Anand Agricultural University, Anand, India

Abstract : Testing is very important phase of software development. Stress testing is a form of deliberately intense or thorough testing used to determine the stability of a given system or entity. It involves testing beyond normal operational capacity, often to a breaking point, in order to observe the results. Stress testing examines how the system behaves under intense loads, and how it recovers when going back to normal usage. JMeter is a testing tool for stress testing. This study aims to carry out stress testing of different websites, to record different parameters and analyze them.

IndexTerms **- Apache JMeter, Stress Testing.**

I. INTRODUCTION

Stress testing is used to test the stability and reliability of the system. This test mainly determines the system on its robustness and error handling under extremely heavy load conditions. It even tests beyond the normal operating point and evaluates how the system works under those extreme conditions. Stress testing is done to make sure that the system would not crash under crunch situations. Stress testing is also known as endurance testing. Under stress testing, application under test is to be stressed for a short period of time to know its withstanding capacity. Most prominent use of stress testing is to determine the limit, at which the system or software or hardware breaks. It also checks whether system demonstrates effective error management under extreme conditions.[1] *Stress testing* is a type of non-functional testing. Under this, various activities to overload the existing resources with excess jobs are carried out in an attempt to break the system down. The purpose behind *stress testing* is to ascertain the failure of system and to monitor how the system recovers back gracefully. If the system is able to recover without loss of data and without creating security leaks, it means that it has successfully passed the stress test.[4]

Stress testing can be conducted through load testing tools, by defining a test case with a very high number of concurrent virtual users. If your stress test includes a sudden ramp-up in the number of virtual users, it is called **a** Spike Test. If you stress test for a long period of time to check the system's sustainability over time with a slow ramp-up, it's called a Soak Test.[2]

The Apache JMeter application is open source software, a 100% pure Java application designed to load test functional behavior and measure performance. It may be used to test performance both on static and dynamic resources, Web dynamic applications. It can be used to simulate a heavy load on a server, group of servers, network or object to test its strength or to analyze overall performance under different load types.[3]

JMeter send requests to a target server by simulating a group of users then collect data to calculate statistics and display performance metrics through various formats. JMeter workflow diagram is depicted as under.[4]

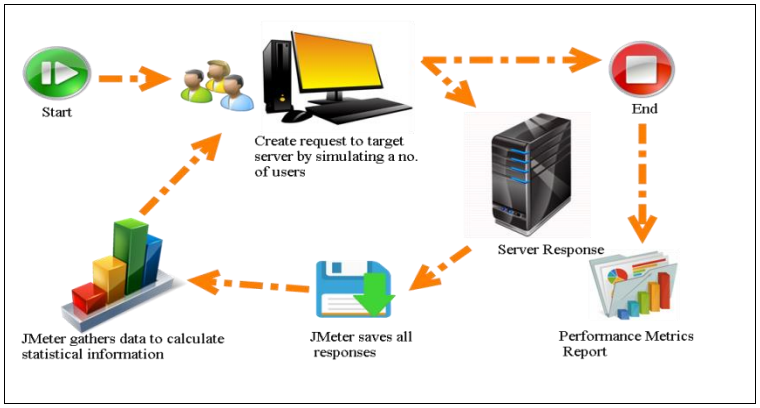


Fig 1. JMeter workflow diagram

II. MATERIALS AND METHODS

Three different Websites are selected for this experiment.

1. Society of Extension Education, Gujarat Website (www.gjoee.org)
2. Google Website (www.google.com)
3. Yahoo Website (www.yahoo.com)

The home page of the website is access using same load for all websites. JMeter testing tool is used to generate different user load and http requests. All http requests are generated from the same machine. The machine configuration is as under.

Machine Configuration

Type: Desktop
Processor: AMD Phenom II X# 720 Processor 2.80 GHz
OS: Windows 7 Enterprise(32-bit)
RAM: 3.25 GB
Apache JMeter: Version 3.2r1790748

The diagram given as below depicts the set up for the experiment.

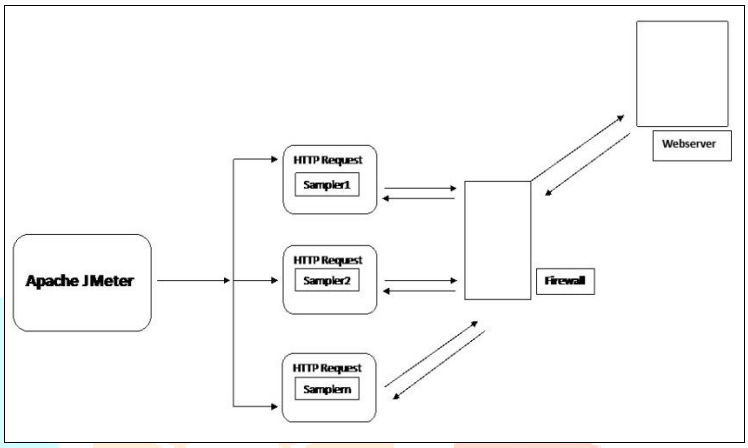


Fig 2. Experiment Setup

The experiment will generate different performance parameters like Sample Time(Turn around time), Latency and Throughput etc. which are recorded and analyzed. The experiment is carried out in Non GUI mode as suggested in the JMeter documentation for better results.

Parameters in Apache JMeter

Test Plan	No. of Threads(User)	Ramp up Period (in seconds)	Loop count	Http Sampler request
Yahootestplan.jmx	10,20,30,40,50	10	1	http://www.yahoo.com
Googletestplan.jmx	10,20,30,40,50	10	1	http://www.google.com
Gjoeetestplan.jmx	10,20,30,40,50	10	1	http://www.gjoee.org

Important performance parameters measured in this experiment are defined as below.

- **Latency**: The number of milliseconds that elapsed between when JMeter sent the request and when an initial response was received.
- **Sample Time**: The number of milliseconds that the server took to fully serve the request (response + latency).
- **Throughput** is calculated as requests/unit of time. The time is calculated from the start of the first sample to the end of the last sample. This includes any intervals between samples, as it is supposed to represent the load on the server. The formula is: Throughput = (number of requests) / (total time). The **Throughput** is the most important parameter. It represents the ability of the server to handle heavy load. The **higher** the Throughput is, the **better** is the server performance.
- **Connect Time**. JMeter measures the time it took to establish the connection, including SSL handshake. Note that connect time is not automatically subtracted from latency. In case of connection error, the metric will be equal to the time it took to face the error, for example in case of Timeout, it should be equal to connection timeout.
- The **deviation** indicates the deviation from the average. The **smaller** the **better**.

III. Results and Discussion

Parameters	**10_10**	**20_10**	**30_10**	**40_10**	**50_10**
Sample	10	20	30	40	50
Sample time (Avg.)	273ms	336ms	376ms	262ms	256ms
Sample time	278ms	270ms	245ms	247ms	239ms

(Median)					
Sample time (Deviation)	36ms	155ms	542ms	46ms	62ms
Throughput	65.782/Min	126.117/Min	185.931/Min	242.718/Min	303.337/Min
Latency	117.4ms	175.95ms	220.63ms	110.77ms	108.94ms
Connect time	12.6ms	13.3ms	21.76ms	7.15ms	11.34ms
Error	0.0%	0.0%	0.0%	0.0%	0.0%
Byte send	0.25KB/Sec	0.48KB/Sec	0.70 KB/Sec	1.83 KB/Sec	1.15 KB/Sec
Byte received	16.46KB/Sec	29.19KB/Sec	43.18KB/Sec	112.43 KB/Sec	70.25 KB/Sec

Table 1: Results for Google home page with different number of requests per second

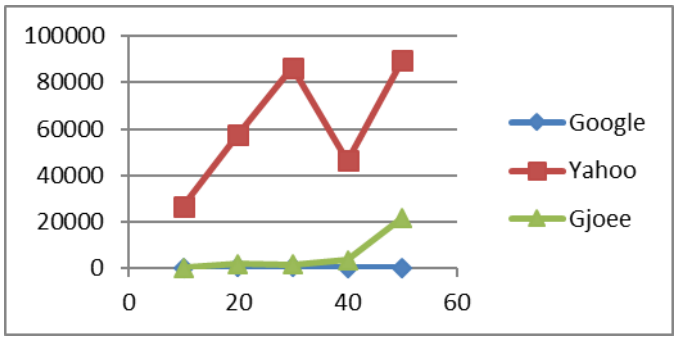
Parameters	**10_10**	**20_10**	**30_10**	**40_10**	**50_10**
Sample	10	20	30	40	50
Sample time (Avg.)	26600ms	57664ms	86337ms	46456ms	89715ms
Sample time (Median)	25891ms	58052ms	67670ms	57460ms	93658ms
Sample time (Deviation)	4655ms	7371ms	169881ms	29208ms	39275ms
Throughput	15.143/Min	15.968/Min	0.011/Min	23.822/Min	15.787/Min
Latency	1489.4ms	1936.2ms	987.70ms	1054.7ms	1612.5ms
Connect time	321.1ms	1368.65ms	15.33ms	215.5ms	432.56ms
Error	0.0%	0.0%	19.35%	45.00%	26.00%
Byte send	0.09KB/Sec	0.09KB/Sec	0.00 KB/Sec	0.29 KB/Sec	0.08 KB/Sec
Byte received	96.06KB/Sec	103.22KB/Sec	0.06KB/Sec	267.92 KB/Sec	80.26 KB/Sec

Table 2: Results for Yahoo home page with different number of requests per second

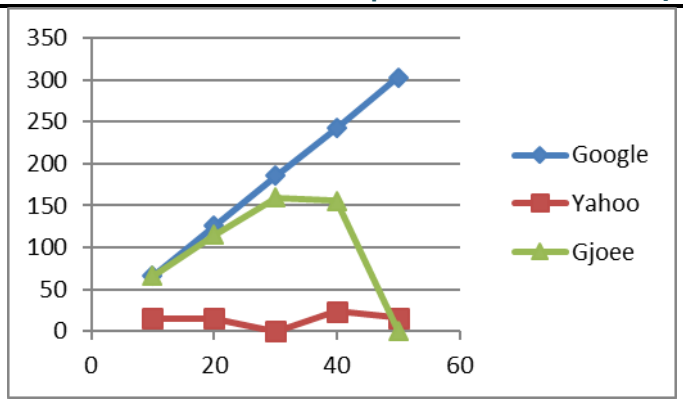
Parameters	**10_10**	**20_10**	**30_10**	**40_10**	**50_10**
Sample	10	20	30	40	50
Sample time (Avg.)	273ms	1756ms	1435ms	3396ms	21616ms
Sample time (Median)	278ms	1125ms	1503ms	2655ms	22253ms
Sample time (Deviation)	36ms	815ms	250ms	1714ms	17226ms
Throughput	65.782/Min	115.086/Min	159.631/Min	155.935/Min	0.689/Min
Latency	1132.4ms	1497.05ms	1110.767ms	2572.5ms	13520.74ms
Connect time	25.5ms	12.4ms	5.93ms	12.125ms	11.12ms
Error	0.0%	0.0%	0.0%	0.0%	0.0%
Byte send	0.00KB/Sec	0.22KB/Sec	0.30 KB/Sec	0.92 KB/Sec	0.00 KB/Sec
Byte received	0.00KB/Sec	69.03KB/Sec	95.74KB/Sec	56.22 KB/Sec	0.00 KB/Sec

Table 3: Results for Gjoee home page with different number of requests per second

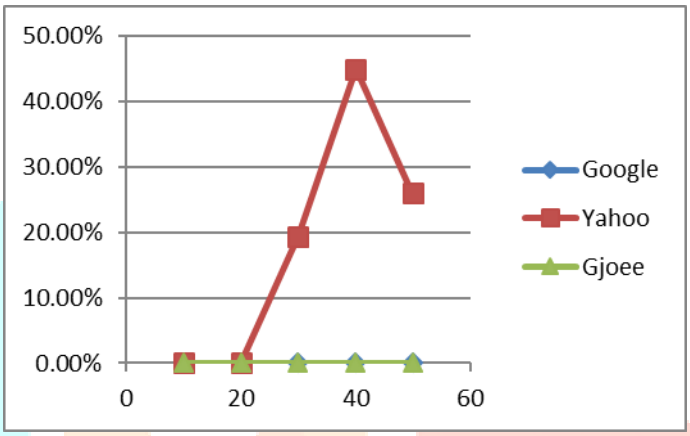
The results of the experiment show following important points.



Graph 1: Sample Time(Avg.) in ms



Graph 2: Throughput/Min



Graph 3: Error(%)

1. As shown in Graph 1, Sample Time(Avg.) remains consistent in case of Google. There is some variation in case of Gjoee website. While results of Yahoo are inconsistent.
2. As shown in Graph 2, Throughput of Google has been increasing linearly with the increasing number of requests. This is very good property of a website to handle pick load on the website. It remains constant in the case of Yahoo. Results show that site in poorly scalable with increasing number of requests. While results of Gjoee are inconsistent.
3. As shown in Graph 3, Error rate(%) is zero in case of Google and Gjoee while it is high and inconsistent in case of Yahoo.

IV. CONCLUSION

The experiment carried out in this paper tries to show that stress testing can be easily done with open-source testing tool like JMeter. Three websites are tested for the different Stress level that is different number of web requests per second. The experiment is carried out only once but can be repeated for N number of times to increase reliability of the results.

REFERENCES

[1] https://www.guru99.com/stress-testing-tutorial.html
[2] https://www.blazemeter.com/blog/performance-testing-vs-load-testing-vs-stress-testing
[3] http://jmeter.apache.org/
[4] http://toolsqa.com/jmeter/what-is-apache-jmeter/
[5] Yogesh R. Ghodasara, R S Parmar, Gautam J Kamani, Krunal C Kamani(2018). *Performance Testing of Websites Using JMeter Testing Tool*. International Journal of Creative Research Thoughts(IJCRT), 6(1),268-271.