

PERFORMANCE TESTING OF WEBSITES USING JMETER TESTING TOOL

¹Yogesh R. Ghodasara, ²R S Parmar, ³Gautam J Kamani, ⁴Krunal C Kamani

¹Professor, ²Professor, ³Asst. Professor, ⁴Asst. Professor

¹Department of Agricultural Information Technology,

¹Anand Agricultural University, Anand 388 110, India

Abstract: Testing is very important phase of software development. Software testing is the process of validating and verifying the correctness of software. Performance testing is used to determine the responsiveness, throughput, reliability, and/or scalability of a system under a given workload. JMeter is a testing tool for performance testing. This study aims to carry out performance testing of different websites, record different performance parameters and analyze them.

IndexTerms - Apache JMeter, Performance Testing

I. INTRODUCTION

Performance testing is a type of testing to ensure software applications will perform well under their expected workload. Features and Functionality supported by a software system is not the only concern. A software application's performance like its response time, reliability, resource usage and scalability do matter. The goal of performance testing is not to find bugs but to eliminate performance bottlenecks. The focus of Performance testing is checking a software program's Speed, Scalability and Stability[1].

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[2]. JMeter is useful tool to improve software product quality[3]. Apache JMeter may be used to test performance both on static and dynamic resources. It can be used to simulate a heavy load on a server, network or object to test its strength or to analyze overall performance under different load types[4].

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[5].

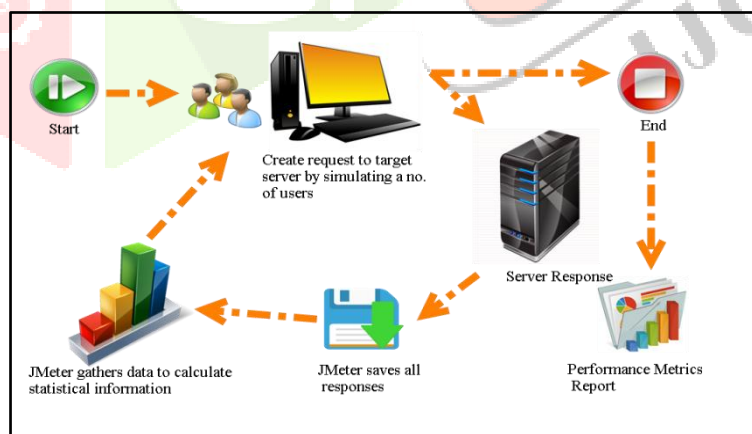


Figure 1: JMeter workflow diagram

II. MATERIALS AND METHODS

Three different Websites are selected for this experiment.

1. Society of Extension Education, Gujarat Website (www.gjoe.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.

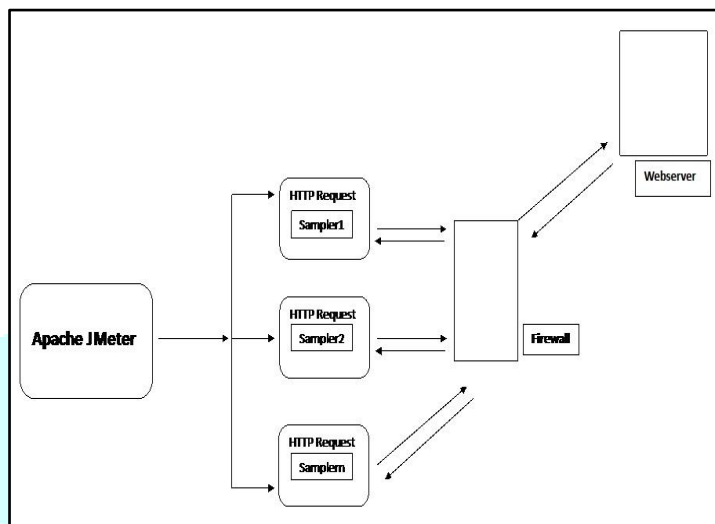


Figure 2: experiment setup

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

Table 1: Test plan summary

Test Plan	No. of Threads(User)	Ramp up Period (in seconds)	Loop count	Http Sampler request
Yahootestplan.jmx	10	10	1	http://www.yahoo.com
Googletestplan.jmx	10	10	1	http://www.google.com
Gjoetestplan.jmx	10	10	1	http://www.gjoe.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: $\text{Throughput} = \frac{\text{number of requests}}{\text{total time}}$. 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

Table 2: Result summary

Test Plan	Yahootestplan.jmx	Googletestplan.jmx	Gjoeetestplan.jmx
Sample	10	10	10
Byte send	351	276	113
Byte received	395294	14613	36507
Sample time(Avg.)	23631 ms	1095 ms	2640 ms
Sample time(Median)	23042	339 ms	2670 ms
Sample time(Deviation)	5007 ms	1062 ms	1415 ms
Throughput	16.295/Min	53.976/Min	51.975/Min
Latency	1794.1 ms	396.9 ms	2636.7 ms
Connect time	913.5 ms	23.8 ms	13.6 ms
Error	0.0%	0.0%	0.0%
Byte send	0.09 KB/Sec	0.24 KB/Sec	0.10 KB/Sec
Byte received	104.84 KB/Sec	12.84 KB/Sec	30.88 KB/Sec

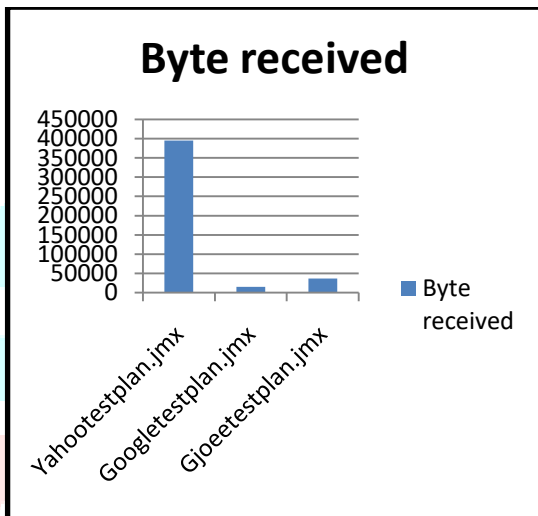


Figure 3: Byte received

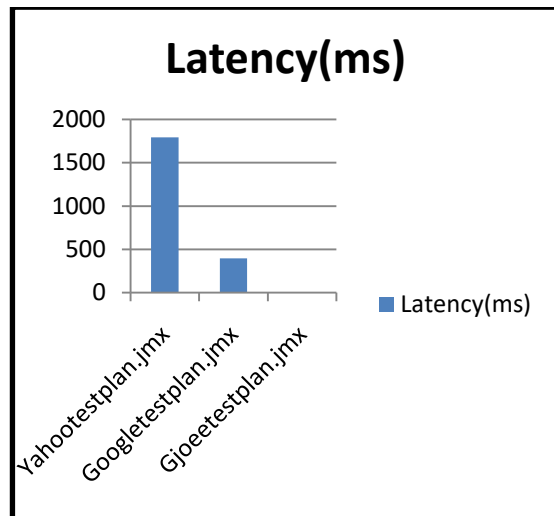


Figure 4: Latency

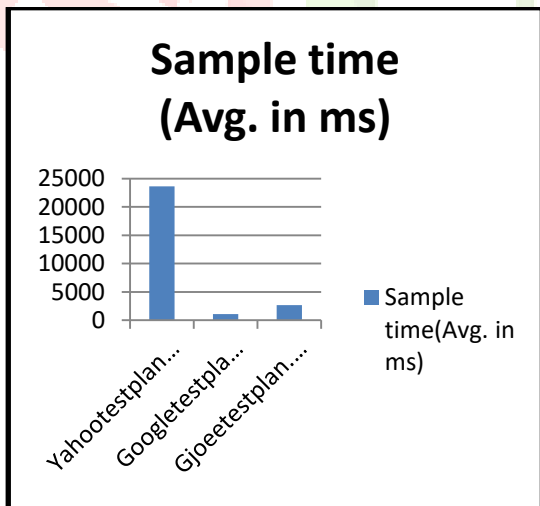


Figure 5: Sample time

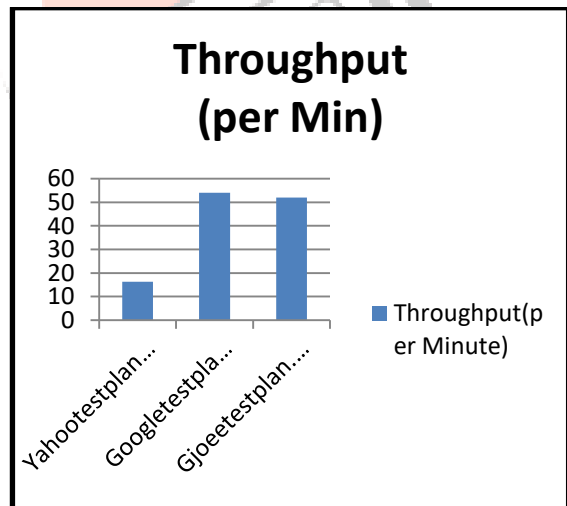


Figure 6: Throughput

The results of the experiment show following important points.

- Byte received is highest in the case of Yahootestplan. It means the page is highly loaded. Performance can be improved by making page light weight.
- Latency is highest in case of Gjoeetestplan while lowest in Googletestplan. Lower value shows better network connectivity available.
- Average Sample Time is lowest in Googletestplan which reflects better turnaround time for web request and leads to better website performance.

- Googletestplan gives highest throughput. It serves 54 requests per minute which is 17 in case of Yahooetestplan and 52 in case of Gjoetestplan.
- Error rate is zero in all three cases. This parameter shows that all websites are capable to handle given load.

IV. CONCLUSION

JMeter can be a very valuable tool for determining how your web application server setup should be improved, to reduce bottlenecks and increase performance. JMeter can be used to create new test plans to measure the performance of your servers in various scenarios. The test that we used as the example does not accurately reflect a normal user's usage pattern, but JMeter has the tools to perform a variety of tests that may be useful in your own environment. For example, JMeter can be configured to simulate a user logging into your application, client-side caching, and handling user sessions with URL rewriting. There are many other built-in samplers, listeners, and configuration tools that can help you build your desired scenario.

REFERENCES

- [1] <https://www.guru99.com/performance-testing.html>
- [2] <http://jmeter.apache.org/>
- [3] <http://ijarcsms.com/docs/paper/volume2/issue1/V2I1-0026.pdf>
- [4] <https://www.ijccr.com/May2013/3.pdf>
- [5] <http://toolsqa.com/jmeter/what-is-apache-jmeter/>

