



## Research on End to End Performance Testing of Web Applications

Lalit Kumar Garg, Preeti Rani (Students), Deepika Goyal (Guide)

Department of Computer Science and Engineering, Advance Institute of technology & Management  
Affiliated to M D University Rohtak, India

DOI: [10.23956/ijarcsse/V7I2/0120](https://doi.org/10.23956/ijarcsse/V7I2/0120)

**Abstract**—In this paper, we are planning to understand end to end performance testing process along with the relevant tools. In this Cycle we would cover brief idea about the major steps performed during performance testing and monitoring starting from test environment to result analysis, based on the Approach we will recommend and give a scope for future research. Main focus is to study about the various types of Testing tools that are used to load test the application and server performance monitoring, Importance of Performance testing in Web applications. We will study all the areas/components that will arise a need of End to End Performance testing framework.

**Keywords**- load testing, stress testing, load testing tool, performance testing, server monitoring, and load test scripting. Server hardware resource monitoring

### I. INTRODUCTION

Performance testing is not just about front end OR back end testing. Understanding how the application will perform is more than that. It is about knowing how the application will behave end-to-end, which requires engineering the application performance. For that reason, it is important to learn the rationality how to design, execute and understand the results based on the business requirements.

Majorly performance testing includes Load Testing, Server Side monitoring, Bottleneck Identifications, Server Tuning as well as Revalidation executions.

**Performance testing.** This type of testing determines or validates the speed, scalability, and/or stability characteristics of the system or application under test. Performance is concerned with achieving response times, throughput, and resource-utilization levels that meet the performance objectives for the project or product. In this guide, performance testing represents the superset of all of the other subcategories of performance-related testing.

Performance testing is typically done to help identify bottlenecks in a system, establish a baseline for future testing, support a performance tuning effort, determine compliance with performance goals and requirements, and/or collect other performance-related data to help stakeholders make informed decisions related to the overall quality of the application being tested. In addition, the results from performance testing and analysis can help you to estimate the hardware configuration required to support the application(s) when you “go live” to production operation

### Activities of Performance Testing

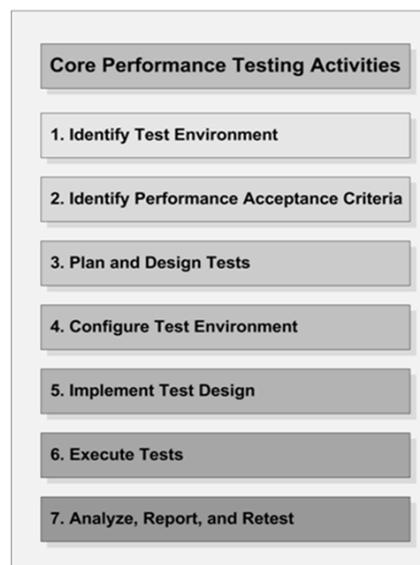


Fig 1.0 – Performance Testing Activities

## Types of Performance Testing

**Load test:** To verify application behaviour under normal and peak load conditions.

**Stress test:** To determine or validate an application's behaviour when it is pushed beyond normal or peak load conditions.

**Capacity test:** To determine how many users and/or transactions a given system will support and still meet performance goals.

## II. REARCH & DISCUSSIONS

To generate the load on the servers, Virtual users are triggered from the automated tools called as load testing tools.

During the load generation servers performance related parameter like CPU, Memory , IO, Networks are monitored to see the impact of load and to benchmark the corresponding application. Server side monitoring can be done either using the inbuilt utilities of Windows or Linux servers or some tool/APS are also available in the marketing. Below are some of the popular load testing tools in the market.

### Load Testing Tools:

- WebLOAD
- LoadComplete
- Apache JMeter
- LoadRunner
- Appvance
- NeoLoad
- LoadUI
- WAPT
- Loadster
- LoadImpact
- Rational Performance Tester
- Testing Anywhere
- OpenSTA

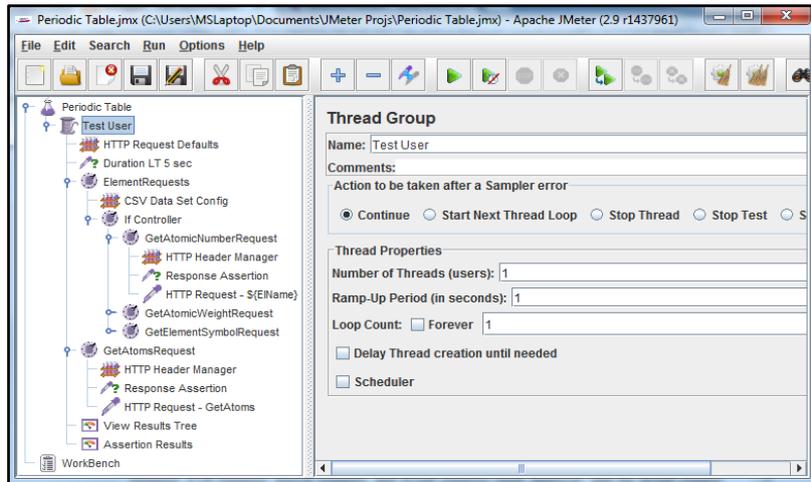


Fig 1.1 – Apache JMeter

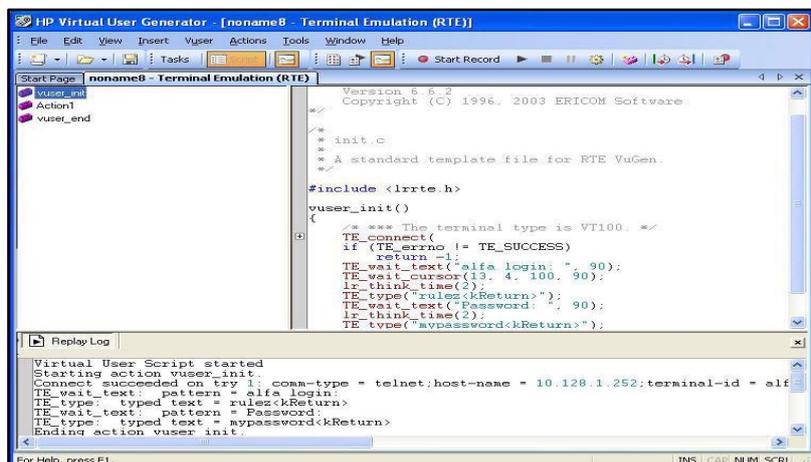


Fig 1.2 – HP Load Runner

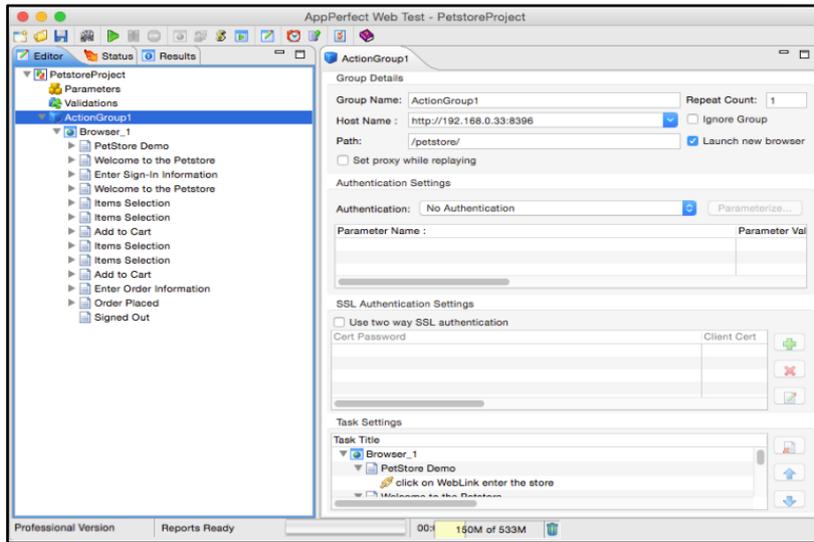


Fig 1.3 – Webload

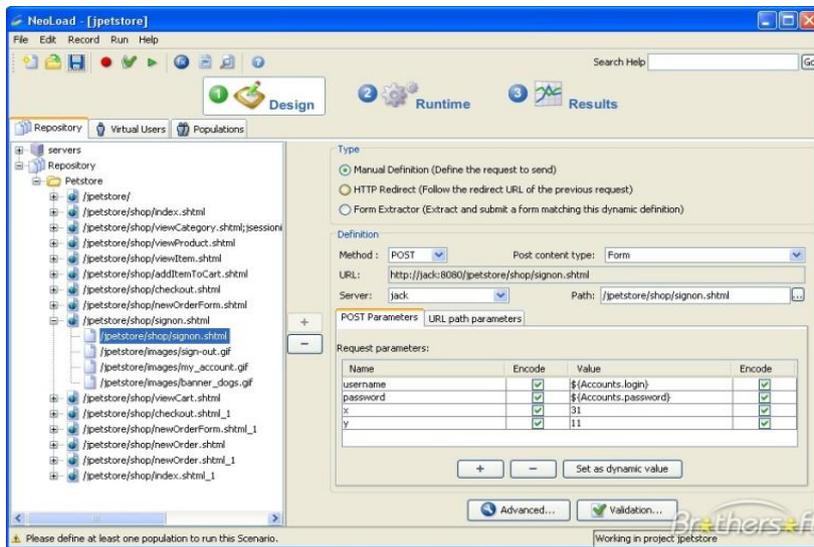


Fig 1.4 – Webload

### Server Monitoring Tools:

Windows servers can be monitored using windows inbuilt utility called as “Perfmon”

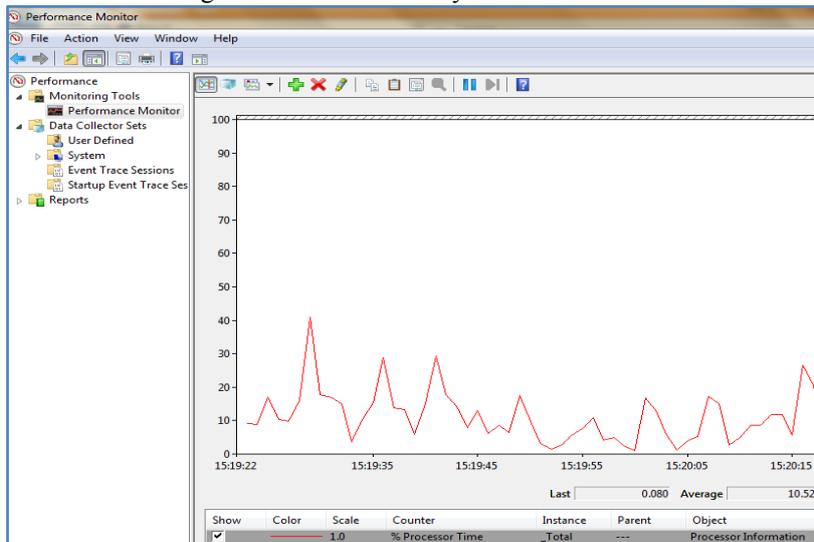


Fig 1.5 – Perfmon Counters

Linux Servers can be monitored using the commands (After the installations)

```
sh-4.3$ echo 'System is initializing...please wait'
System is initializing...please wait
sh-4.3$ /bin/bash
bash-4.3$ vmstat
procs-----memory-----swap-- ----io---- -system-- -----cpu-----
 r b swpd free buff cache si so bi bo in cs us sy id wa st
 4 1 2954220 119828240 1860 54793420 18 41 634 210 0 0 5 3 91 0 0
bash-4.3$
```

Fig 1.6 – VMSTST

```
top - 10:23:18 up 8 days, 20:34, 0 users, load average: 8.42, 19.64, 21.58
Tasks: 5 total, 1 running, 3 sleeping, 0 stopped, 1 zombie
%Cpu(s): 7.9 us, 4.9 sy, 0.0 ni, 86.6 id, 0.4 wa, 0.0 hi, 0.1 si, 0.0 st
KiB Mem : 26386318+total, 12183630+free, 87577544 used, 54449340 buff/cache
KiB Swap: 8101884 total, 5152584 free, 2949300 used. 16747667+avail Mem

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
 1 22474 20 0 1121920 78888 7380 S 0.0 0.0 0:01.91 --CODINGGROUND-
29 22474 20 0 0 0 0 Z 0.0 0.0 0:00.00 bash
33 22474 20 0 16132 1920 1536 S 0.0 0.0 0:00.00 sh
36 22474 20 0 16132 1932 1548 S 0.0 0.0 0:00.00 bash
40 22474 20 0 54884 2044 1504 R 0.0 0.0 0:00.00 top
```

Fig 1.7 – TOP

```
(marsl@kU) ~ ->
$ dstat
-----total-cpu-usage-----disk-total-----load-avg-----memory-usage-----net-total-----procs-----swap-----system-----
usr sys idl wat hqg sig read writ in 5m 15m used buff cach free recv send run blk new used free int csw
13 3 84 1 0 0 75% 148 0.39 0.53 0.75 908M 168M 1454M 1397M 0 0 0 0 1.4 0 3986M 969 3890
6 3 91 1 0 0 0 40 0.39 0.53 0.75 908M 168M 1459M 1391M 0 0 1.0 0 0 0 3986M 705 2974
2 1 97 0 0 1 0 0 0.39 0.53 0.75 908M 168M 1455M 1396M 0 0 1.0 0 0 0 3986M 554 1193
3 1 96 0 0 0 0 0 0.43 0.54 0.75 908M 168M 1461M 1389M 0 0 0 0 0 0 3986M 585 1518
2 0 97 0 0 0 0 0 0.43 0.54 0.75 908M 168M 1461M 1389M 0 0 1.0 1.0 0 0 3986M 603 1351
2 1 97 0 0 0 0 0 0.43 0.54 0.75 908M 168M 1462M 1389M 0 0 0 0 0 0 3986M 657 1365
3 1 96 0 0 0 0 16 0.43 0.54 0.75 908M 168M 1462M 1388M 0 0 1.0 0 0 0 3986M 777 1599
3 1 96 0 0 0 0 0 0.43 0.54 0.75 909M 168M 1463M 1387M 0 0 1.0 0 0 0 3986M 864 1829
```

Fig 1.8 DSTAT

```
[root@server ~]# iostat -N
Linux 2.6.32-279.el6.x86_64 (server.autel.com) 09/06/2013 _x86_64_
CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.17    0.00    0.59    0.57    0.00   98.67

Device:            tps    Blk_read/s    Blk_wrtn/s    Blk_read    Blk_wrtn
sda                 2.39         195.79         8.30     693326     29408
sdb                 0.08          0.65          0.00         2312         0
sdc                 0.08          0.65          0.00         2312         0
sdd                 0.08          0.65          0.00         2312         0

[root@server ~]#
```

Fig 1.9 IOSTAT

Application performance management (APM) is the monitoring and management of performance

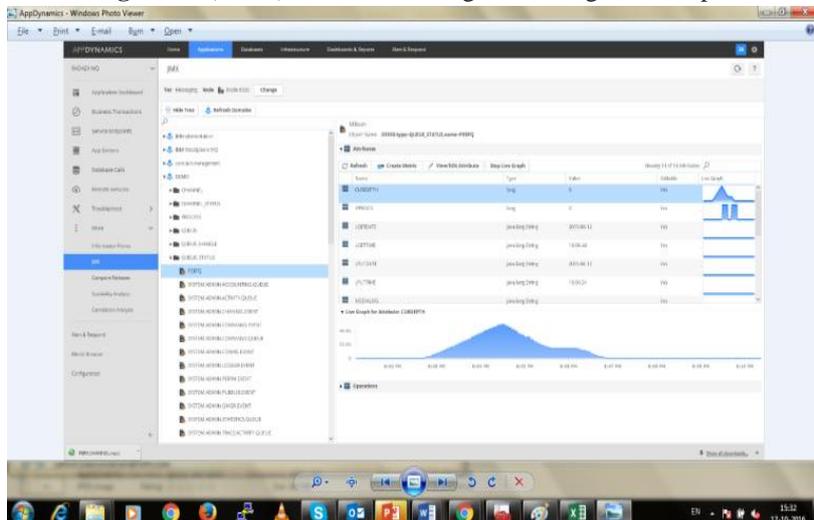


Fig 1.10 Appdynamics

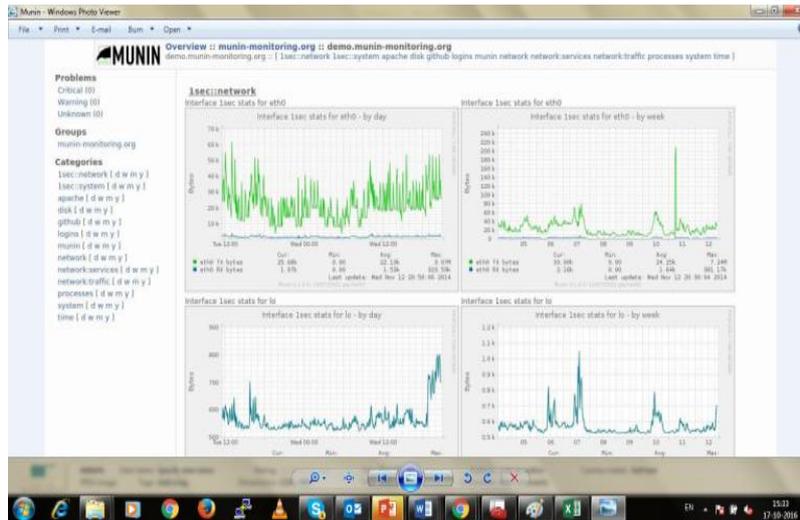


Fig 1.11 MUNIN

In the best case, a team with the technical expertise on the load testing tool is deployed for an assignment for script preparation and load test executions to generate load on the server.

At the same time resources with expertise in the server side monitoring with the performance monitoring tools/utilities/APM are involved at the backed.

After the compilations of results from the load test tools and corresponding results of monitoring tools, Issues are identified and then recommendations/suggestions will be sent to the concern team by result analyst.

Development team will fix the issues reported by the testing team and then revalidation comparative executions will be performed to see the impact of fixes on the application.

These issues and their fixes can be on application side, database side and hardware side.

### III. CONCLUSION

Study was conducted to understand the scope of the need of a framework that can build over all the load testing tools in back end and serve with the performance monitoring capabilities also.

Managing the end to end performance testing is a complex mechanism that involves the deployment of combinations of man power with tool/ domain specific technical capabilities, as in today's scenario, Application environment is become more and more complex to adhere the need of huge customer base like in case of ecommerce eg – Snapdeal, Flipkart , Paytm etc.

Ever tools is capable to doing the testing of particular type of application/ protocol.

Future work can be done to analyses the feasibility this framework which can generate the load with the combination of open source tools and can also monitor the servers along with various application architectural components.

### REFERENCES

- [1] <http://www.softwaretestinghelp.com/what-is-end-to-end-testing/>
- [2] <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC61333/>
- [3] <http://www.oreilly.com/pub/e/3401>
- [4] <https://msdn.microsoft.com/en-us/library/bb924356.aspx>.
- [5] <http://www.softwaretestinghelp.com/performance-testing-tools-load-testing-tools/>