

Load Runner: Methodology to Performance Testing

Ashish Kumar

Performance Engineer, Tata Consultancy Services Ltd, Mumbai, Maharashtra, India

Email: ashish.ksinha@yahoo.co.in / ashish.k40@tcs.com

ABSTRACT

The paper covers the most important part of testing which is essential for testing the performance of application before going 'Live'. In my research and analysis with Microfocus owned tool 'Load Runner', we will discuss about it deals with web based application and approach to calculate transaction per hour (TPH) for test execution. As tool supports many protocols based on the nature of application. Performance testing is used to analyze the real time response time for business transaction. Application be constant with increasing load or with simultaneous users should not affect the performance of the application, is our main motto as a Performance Tester/Engineer.

KEYWORDS: Performance Testing, Load Runner, Performance Center, Virtual Generator, Transaction, Load Runner Analysis

How to cite this paper: Ashish Kumar "Load Runner: Methodology to Performance Testing"

Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-4 | Issue-2, February 2020, pp.878-880, URL: www.ijtsrd.com/papers/ijtsrd30155.pdf



Copyright © 2019 by author(s) and International Journal of Trend in Scientific Research and Development Journal. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (CC BY 4.0) (<http://creativecommons.org/licenses/by/4.0>)



I. INTRODUCTION

Application Performance Testing is all about the performance and quality of the application. The performance testing will insure the behavior of application under their expected workload. Performance testing is a type of Non-functional testing, which simplifies that these testing has no relation with the functionality of the application. The performance testing is done for a web application for its speed, scalability and stability (S³). The goal of performance testing is not to find bug and fix them; it is moreover for elimination of performance bottlenecks.

Performance testing come in light when some unexpected cause start arising even after Manual or functional testing. Many companies loses its credibility as the unexpected behavior of application come in picture in production environment when sudden increase in real time user, end user, back office or admin hits the application. To overcome this big challenge, Performance testing play a vital role. It is a tool based testing where the tool records the events of request and re-play it with simulation of different users, just say real time virtual user.

II. TYPES OF SOFTWARE TESTING

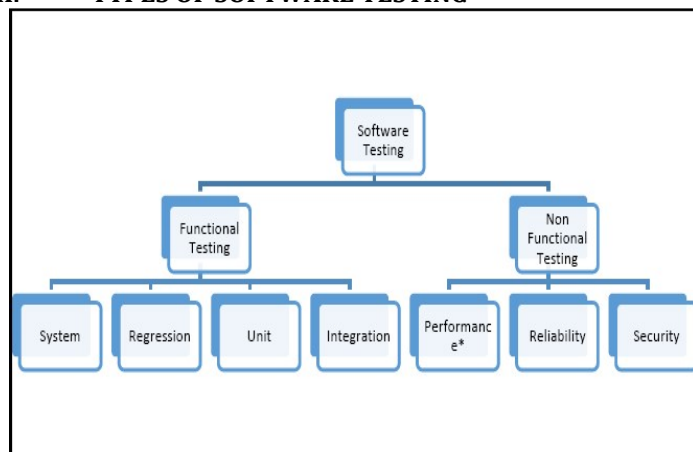


Fig1: Types of Software Testing

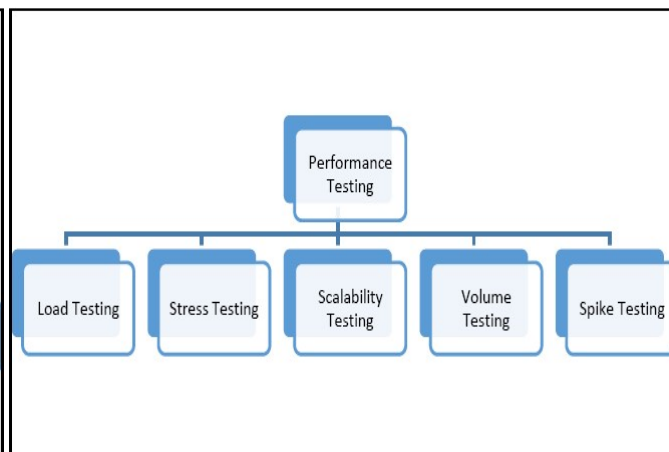


Fig2: Types of Performance Testing

- **Load Testing** – In load, testing application has been checked under various load to test any bottleneck of the application.

- **Stress Testing** – In stress testing, web application is checked under extreme workload to see the breaking point of an application.
- **Scalability Testing** – It is used to check the load with increasing number of user of an application, and it is helpful in increasing the capacity of software system.
- **Volume Testing** – In volume testing, large number of data is stored in database and overall web application behavior is observed. It checks the correctness of an application under varying database volume.
- **Spike Testing** – When there is sudden large spikes in the load generated by users, then spike testing is done.

III. Software Testing Tool – Load Runner

Load Runner is software-testing tool owned by Micro Focus. Load Runner is a best tool as per market survey, which simulates user activity by generating messages between application components rather than simulating interactions with the user interface such as keypresses or mouse movements.

HP acquired Load Runner as part of its acquisition of Mercury Interactive in November 2006. In Sept 2016, Hewlett Packard Enterprise announced it is selling its software business, including Mercury products, to Micro Focus. As of 01-Sept-2017, the acquisition was complete.

Components of Load Runner

- Virtual Generator
- Analysis
- Controller
- Load Agents
- Agents Process
- Recording

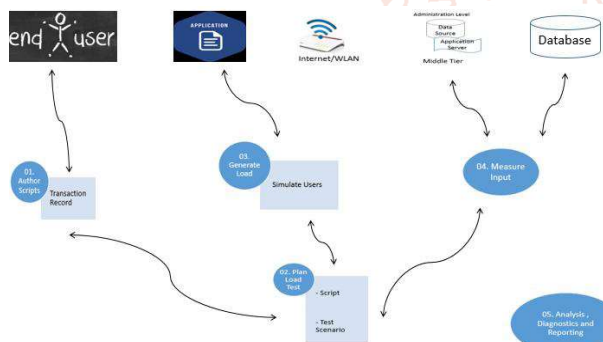


Fig3: Components of Load Runner

When you start the recording in Vugen for a new application. Record the same business flow twice that will help in comparing both vugen scripts and an easy way to find out the dynamic values. Once we have a list of dynamic values, we can easily start correlation. Correlation in Load Runner is defined in two ways as automatic and manual correlation.

Suppose we had a dynamic value 'ABC001', when you are validating the same in code generation, you will have Left and Right boundary e.g. 78_LR_ABC001-XX. In this example, 70_LR_ will be left boundary and _XX will be right boundary. Any dynamic value in this boundary will be captured through tool.

➤ Why Load Runner?

Traditional way of testing approach is known as Manual testing which offers partial solution to load testing. When Client/Product owner wants to ensure that what load their application can sustain, load based testing came into picture.

In addition, to reduce the high count of engineers to perform manual testing, Load testing gives the entire solution for performing this type of testing.

How Loadrunner works?

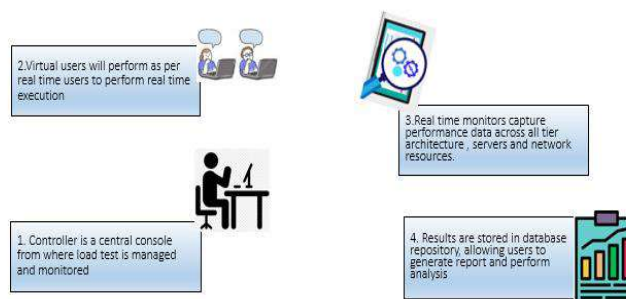


Fig4: How Load Runner works

IV. Automated Performance Testing

With the direct approach to perform the test using Load Runner, you can optimize resources, best suggestion for hardware configuration, and the best network requirement based on application architecture, and debugging the transaction response time based on agreed service level agreement (SLA's).

➤ Planning & Identification of Key business Area in Performance Testing

A user requirement is translated to performance objective for observing realistic behavior of application. In planning and drafting the performance test plan, we need to understand the key business area and their impact on complete business flow of particular LOB based application. Test plan need to complete with the current and projected user specified objectives, which comprises the current statistics of production. For designing the plan, we must understand the architecture of application and we should have strong knowledge to compare the production details for apple-to-apple comparison.

➤ Defining the Data

You need to define what data is required onwards to perform the test. The data can be created dynamically, or making a clone of master user, or writing few interesting creation query in Database. By using realistic data/user, you can create accurate load test.

➤ Defining Testing Approach

You should have depth knowledge in creating the strategy for testing applications. You can choose any type of performance testing, say as Load, Stress, Capacity, Volume, Spike, etc based on the user requirement. This type of testing can be performed with Load Runner.

➤ Making of Performance Scripts

When you are good with the above steps, we need to record, enhance the scripts for completed flow. You need to write `lr_start_transaction()`; before the particular web request and `lr_end_transaction()`; after completion of that request. This function will count the response time of web request for transaction, which will be helpful in comparing the same with agreed SLA's. For enhancing the script, any value you enter via keyboard at the time of recording needs to be parametrize and any dynamic value you observed in code is to be correlated.

➤ Test Scenario and Execution

Once the test script is ready and validated with few iterations (Say 5) and different user, script should be uploaded in Performance center / controller.

We need to make the scenario for our test executions.

- Dedicated Load generator is mapped according to the number of Users which you are going to test.
- While making the scenario, ramp up and ramp down should be given and a steady state of run must be mentioned as per requirement.
- Monitor varieties of graph while running the execution

We need to enable run time settings with our requirement.

1. TPH (Transaction Per Hour)

Little law states the formula as below

$$\text{Number of Users} = \text{TPH} * (\text{Pacing} + \text{Think time}) / 36000$$

2. Logs

When we run the test in performance center, we need to disable the full logs. Why? Because it will write the logs which will require memory. In addition, if your test is of higher virtual users, it will write full logs. So recommended way is to disable the full logs and write the logs only if error occurs.

3. Pacing/Think Time

With the help of little law, calculate the pacing/ think time desired for your execution

Rest other options will be treated as per the project requirement.

➤ Report And Analysis

Once the execution is completed, download Load Runner analysis (.lra) file from PC. Open LRA file in Analysis component of LR. There you can see some pre-defined graphs, and you can select more graph based on your client/project requirement. It is recommended that select some customized report template, and add flavors based on your requirement and generate the report.

V. TRADITIONAL APPROACH TO CALCULATE FORMULA

Little Law:

Little's law named after "John Dutton Conant Little" who was Renault Professor at the MIT. It states the number of request in system (closed) is equal to the product of average number of Requests serviced per unit time and the average time each Request stays in the system.

Numerically,

$$N = (R + \text{ThinkTime}) * X$$

Where,

N = average number of users in a system

X = average throughput or departure rate of users

R = average time spent in the system or response time

VI. EXPERIMENTAL APPROACH

Based on the little law approach, we had

$$\text{Average Number of users} = (\text{Average Time} + \text{ThinkTime}) * \text{Average Throughput}$$

Additionally, we can talk more about simplifying this formula with all other timing required to manipulate the formula is think time, pacing, average response time with 1 iteration, etc

$$\text{No. of users} = \{(\text{Average Response Time in 1 iteration} + \text{Pacing Time} + \text{Think Time}) * \text{TPH}\}$$

On dividing the result with 3600, we will get the response in seconds (TPS). Remember TPH means transaction per hour. The numerical explanation for little law is explained as below in spreadsheet formula

	A	B	C	D	E	F	G	H
	Script completion time including thinktime	Think Time	Pacing	TPH	Number of Users			
1								
2	50	8.7	8	15	1000.5			
3								
4								
5								

Ashish Kumar:
=(A2+B2+C2)*D2

Many performance engineer/tester believes whatever the result they get is accurate, but it does not be!

For example, if you run a Load Runner test of 1000 concurrent users, Load Runner will give some results. Never assume the given result is 100% accurate. Always it is recommended to cross verify the results using the formula stated above.

Let us say, TPH is 50 and Avg. response time including think time is 15 seconds.

$$\text{Number of users} = 50 * 15 = 750$$

However, our expected result was 1000, something is wrong here! So we need to place the count with the formula stated above (refer spreadsheet picture above) to calculate the accurate measurement.

Reference:

- [1] https://en.wikipedia.org/wiki/Load_Runner
- [2] https://www.softwaretestinghelp.com/Load_Runner/Load_Runner-scripting-challenges/
- [3] https://www.softwaretestinghelp.com/Load_Runner/hp-Load_Runner-tutorial-8/
- [4] <https://www.perftesting.co.uk/performance-center-architecture-overview/2011/08/24/>
- [5] https://perfmatrix.blogspot.com/p/hp-Load_Runner.html
- [6] https://www.guru99.com/understanding-vugen-in-Load_Runner.html
- [7] https://www.softwarehour.com/Load_Runner/parameterization-Load_Runner-step-step/
- [8] https://www.tutorialspoint.com/Load_Runner_online_training/Load_Runner_parameterization.asp
- [9] http://www.geekinterview.com/question_details/15651
- [10] <http://nonfunctionaltestingtools.blogspot.com/2015/01/verifications-and-error-handling-in.html>



Mr. Ashish Kumar, an engineering graduate from VelTech University, Chennai in the field of Electronics & Communication Engineering. After completing his degree, he worked in IT companies like HCL, Accenture and currently working

with TCS on the role of Performance Engineer. He has expert knowledge in performance testing tool Load Runner with multiple hands on experience in monitoring tools.

