**HP LoadRunner Tutorial Topics**

1. Performance Testing Basics, Introduction to LoadRunner & Architecture
2. How to download and install LoadRunner for free
3. Introduction to Virtual User Generator – VuGen Tutorial
4. Advanced VuGen Tutorial – Transactions, Rendezvous points, Parameters etc
5. How to use Controller in LoadRunner?
6. How to use Analysis in LoadRunner?
7. What is Correlation? Benefits and Examples
8. HP LoadRunner Certification Exam
9. HP LoadRunner Interview Questions and Answers

**Basics of Performance Testing and Introduction to LoadRunner**

Before we start using LoadRunner, it is important to have a basic understanding of Performance Testing and why we need to test the performance of any application. It is also helpful to have a brief idea of the various modules in LoadRunner.

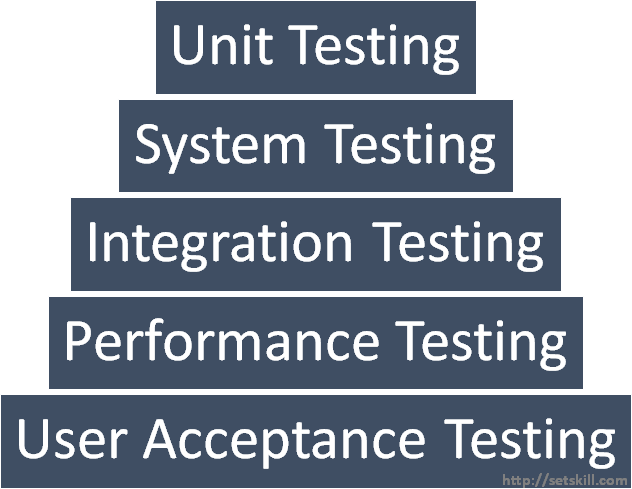
In this article, you will learn

1. What is Performance Testing?
2. Why do need Performance Testing?
3. What is LoadRunner?
4. LoadRunner Architecture
5. Performance Testing Process

**What is Performance Testing?**

Performance Testing is conducted to see how the application behaves under various load conditions. Performance testing can be broken down into the following categories.

* **Load Testing** – This category looks at whether your application meets the Service Level Agreements which have been defined when the system is subject to a particular load. For example, suppose you have a login page for an application. There could be a defined SLA defined saying that this page should always load in 2 seconds. Hence no matter how many users hit the application at one time, this page should always load within that defined time frame. Load testing can also be used to determine the breaking point of your application. So it could determine that for a user load of 100 users, the application just fails to load.
* **Stress Testing** – This is normally conducted to find issues that would prop up whenever there is a high load on the system. For example there can be cases wherein the memory heap size may not be enough when an application is subject to a high load. In such cases certain steps may need to be taken to ensure the application can work under loads.
* **Capacity Testing** – This is used to determine what the future growth requirement for an application is if the load on the application does increase in the future. So questions like “Does the Infrastructure require more memory or CPU to sustain a high load” can be answered during this testing phase.

Performance Testing is normally conducted after System/Integration Testing, in the entire testing life cycle. The entire testing life cycle is depicted in the below diagram.

Once the System has been validated for its functionality during the System and Integration testing phase, performance testing is conducted. This is done before the UAT or User Acceptance Testing phase.

**Why do we need Performance Testing?**

Let’s consider a popular site which I am sure everybody is aware about – Facebook. Let’s look at some of the user statistics for Facebook.

* There are currently over 1.71 billion active users on Facebook on a monthly basis
* There are currently over 1.13 billion active users who log on to Facebook on a daily basis
* There are 1.57 billion mobile active users on Facebook
* There are around 10 million websites which have the Like and Share Buttons for Facebook

The above numbers are just staggering. Even with the above numbers, The Company needs to ensure that normal functionality on Facebook should perform as it should. Would anyone want to wait for 1 hour to log onto Facebook because of the load on the application. The answer is obviously ‘No’. So the only way to ensure that the web site behaves as it should under a heavy load is to ensure that proper performance testing is carried out to replicate the high number of users on the application.

Let’s look at another popular eCommerce site of Amazon. Amazon has nearly 181 million visitors per month on the site. Just Imagine if ever the Amazon site were to go down if the load on the system was not accounted for. Forbes has predicted that Amazon would lose around 66,240 USD per minute if ever the site were to down. This is a staggering loss to the company.

One can just understand how important it is to conduct a performance test for an application to see how it would behave under various load conditions.

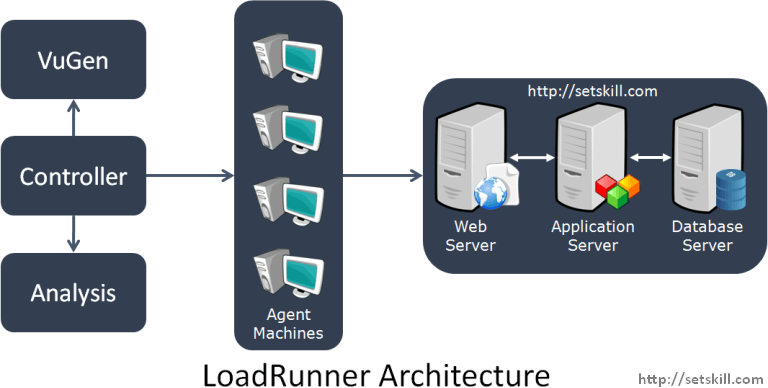
**What is LoadRunner?**

The LoadRunner is a performance testing tool which was initially designed by the company Mercury Interactive. It was later absorbed by Hewlett Packard when the latter acquired Mercury Interactive in November 2006. The most recent version of LoadRunner is 12.53 Patch 1 as per June 2016. LoadRunner is used to simulate the actual user activity on an application. This user activity is multiplied with the help of running virtual users in parallel.

In order to test your application, you need to ensure you have the right protocol purchased along with the LoadRunner tool. So for example, if you wanted to test a web application, you would need to have the Web(HTTP) protocol license. LoadRunner is designed to work with a wide number of protocols. The entire list of protocols is given below.

|  |  |
| --- | --- |
| **Bundle Name** | **Protocol** |
| **.Net record/playback** | Microsoft .NET 1.0, 2.0 and later |
| **Database** | ODBC  Oracle (2-Tier) |
| **DCOM** | Microsoft COM/DCOM |
| **Developer** | Unit Test (NUnit, JUnit, and Selenium)  SDK |
| **Java record/replay** | Java over HTTP Vuser  Java Record/Replay |
| **GUI virtual users** | HPE Unified Functional Testing |
| **Network** | Domain Name Resolution (DNS)  File Transfer Protocol (FTP)  Internet Message Access Protocol (IMAP) Lightweight Directory Access Protocol (LDAP) Microsoft Exchange (MAPI)  Post Office Protocol 3 (POP3)  Simple Mail Transfer Protocol (SMTP)  Tuxedo1 Windows Sockets CORBA  Java RMI  Java (includes ORMI) |
| **Mobile** | Mobile Applications  TruClient—Mobile Web  SAP Mobile Platform (SMP)  Bundle includes equivalent number of HPE Network Virtualization licenses (hp.com/go/nv) |
| **Oracle E-Business** | Oracle NCA  Oracle Web Applications 11i (Click and Script) PeopleSoft Enterprise (Click and Script)  Siebel—Web Web (HTTP/HTML) |
| **Remote access** | Citrix Virtual User (ICA)  Remote Terminal Emulation (RTE) |
| **Remote desktop** | Microsoft Remote Desktop Protocol (RDP) |
| **Rich Internet applications** | AJAX Click and Script  TruClient—Firefox  TruClient—IE  Flex  Silverlight  TruClient—Mobile Web |
| **SAP** | SAP Click and Script  SAP GUI  SAP—Web  SAP Mobile Platform (SMP) |
| **Web 2.0** | Web and multimedia, RIA and SOA bundles |
| **Web and multimedia** | Media Player (MMS)  Real (RealPlayer)  Web (HTTP/HTML)  Mobile Application—HTTP/HTML |
| **Wireless** | Multimedia Messaging Service (MMS) |

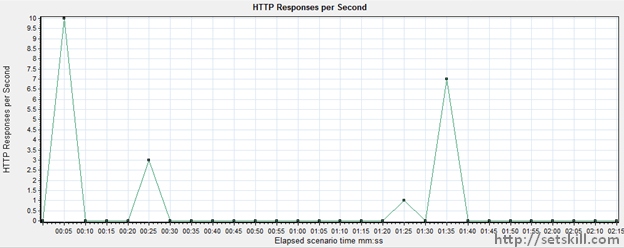
**LoadRunner Architecture**

The LoadRunner architecture is shown below. The key components in LoadRunner are the VuGen (also represented as VUGEN sometimes), the Controller and the Analysis.

Let’s take a look at each component in the LoadRunner architecture. Note that the Application under test can be of any Infrastructure type. Here we are considering a simple web server and a database server.

* **VuGen**– The VuGen component is used to simulate the exact business transaction within your applications. This component generates a script known as Vuser script by recording the user actions. It’s similar to recording macros in Microsoft Excel which can be played back later. Some examples of user steps for an application are given below.
  + Open the login page for the application
  + Log into the application
  + Open a Generate report page
  + Enter the required details to generate the report
  + Generate the Report
* **Controller** – Ideally for any application, you will have many Vuser scripts and various settings under which each script needs to run. The Controller is used to design the load test scenarios and designate how many users will be run against the application.In the Controller you can design the various scenarios which will actually run against the application. You can also use the Controller to start and stop the tests and virtual users which access the application.The Controller can also be used to monitor the performance of each business transaction during the execution of the test.So if we take the above scenarios discussed in VuGen in context, we can have a different mix of users for each user case scenario.
  + **400 users to**Open the login page for the application
  + **350 users to**Log into the application
  + **300 users to** Open a Generate report page
  + **250 users to** Enter the required details to generate the report
  + **100 users to** Generate the Report

With these different mix of scenarios and users, it would give an ideal situation to see how the application behaves under various scenarios and user loads.

* **Agents –**When you execute virtual users, they take up system memory and CPU utilization. Hence it is not ideal for the Controller machine to execute the virtual users.In an ideal scenario, the virtual users are normally executed from separate machines known as Agents. The Controller is used to manage these agents. In this way the performance testing environment is separated by having isolated agent machines which will run the user load.
* **Analysis**– The Analysis component is used to generate results about the Vusers and the test itself. The below graph is an example wherein the number of HTTP Responses per second are shown via a graph.

## Performance Testing Process

Let’s look at the entire process for conducting performance testing. It’s important to design the test cases and know exactly what needs to be carried out for the performance test.

### 1. Stakeholders

Ensure the right stakeholders are in place. In order to run performance tests, all key stakeholders should be in place upfront.

* **Business users –**These are the users who will define the key transactions that need to be included in the performance test.
* **Application Manager –**This is the project manager who is in charge of the application. He is the main stakeholder for any escalations.
* **Test Manager –**This is the person who is responsible for the entire testing process.
* **Test Leads/Test Members –**These are people who will design the VUGEN scripts and executed the load tests.
* **Infrastructure team –**Ensure the entire infrastructure team is place to monitor the servers for the application which is being tested.

### 2. Defining the Business Transactions

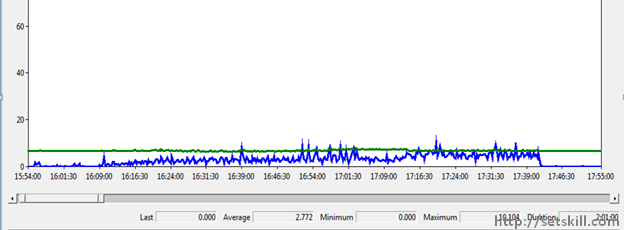
Ideally all business critical transactions should be part of the performance test. Normally the exact test transactions would be defined in the Test Planning document as Non-functional test requirements.

Below is a sample of how the scenarios would be laid out. For each scenario, the concurrent users which need to be fired for a load test and Stress test are given. Secondly the SLA or agreed response time for each scenario is documented.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr no** | **Scenario** | **Load Test – Concurrent Users** | **Stress test -Concurrent Users** | **SLA – Response Time(sec)** |
| 1 | Home | 1 | 4 | 10 |
| 2 | Login | 1 | 4 | 12 |
| 3 | Shared Reports | 1 | 3 | 14 |
| 4 | Dash Board Dataset | 10 | 17 | 20 |
| 5 | Market Data set | 10 | 17 | 22 |
| 6 | Changes Details | 10 | 16 | 30 |
| 7 | Back to Home screen | 5 | 10 | 35 |

### 3. Define the performance Metrics which will be monitored

Apart from the graphs which will be generated by Load runner, you would also need to monitor the servers which are hosting the application. Hence when the load test is running, ensure that the right monitoring tools on the servers are running.

Also ensure the right metrics such as CPU and I/O metrics are being captured as part of the test. Below is an example of performance metrics taken from a Windows based environment for application servers.

### 4. Define the test data

This is a key input. The actual business transactions cannot run if there is no test input. Ensure the application database has the right test data, else the performance tests might give inaccurate results. Also ensure that each test scenario picks up different test data each time it is run. An example is given below

Here we have 2 Virtual users running a scenario. Each virtual user should run with a different user ID so that they can generate varying results. If you run the same scenario for the same user id multiple times, you might get data from a cache, and this might give a wrong indication for the performance test.

* **User A** – Open the login page for the application
* **User A** – Log into the application
* **User A** – Open a Generate report page
* **User A** – Enter the required details to generate the report – **Enters Data A**
* **User A** – Generate the Report
* **User B** – Open the login page for the application
* **User B** – Log into the application
* **User B** – Open a Generate report page
* **User B** – Enter the required details to generate the report – **Enters Data B**
* **User B** – Generate the Report

### 5. Conduct the Performance test

The final task for the testing team is to perform the below steps.

1. Create the VuGen scripts as per the defined business processed.
2. Test the scripts to ensure they run properly.
3. Connect the scripts to the test data.
4. Ensure the test scenarios are defined properly in the Controller.
5. Run the tests during the defined duration.
6. Create the Load test report as per the results generated
7. Log defects from the test in the standard defined defect tracking tool.

Below is a snippet of a load test report which can be generated after a performance test. The table defines the load for 40 and 80 concurrent users. And it gives the server response time, at an average count and 90 percentile count.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Transaction Name** | **Server Response Time in sec** | | | |
| **40 Concurrent Users** | | **80 Concurrent Users** | |
| **Average** | **90 Percent** | **Average** | **90 Percent** |
| Home | 0.25 | 0.47 | 0.441 | 0.905 |
| Login | 0.354 | 0.466 | 0.401 | 0.628 |
| Shared Reports | 0.348 | 0.542 | 0.48 | 0.861 |
| Dash Board Dataset | 4.42 | 5.319 | 5.105 | 6.427 |
| Market Data set | 4.658 | 5.574 | 5.305 | 6.458 |
| Changes Details | 1.678 | 2.177 | 1.864 | 2.496 |
| Back to Home screen | 0.674 | 0.695 | 0.261 | 0.428 |
| Logout | 0.452 | 0.487 | 0.149 | 0.21 |

## Frequently Asked Interview Questions

1. Should performance tests be carried out against production environments?  
   **Answer :** Ideally no. The performance tests should be conducted against environments that are identical to production. The dangers of running performance tests against production environments is that the production environments can go down during a performance test.
2. Should each and every application undergo performance testing?  
   **Answer :** Not necessarily. First of all, only applications which have some sort of client server architecture need to undergo a load test. Secondly only if you are expecting a concurrent user load of 10 or more users should you ideally conduct a load test.