

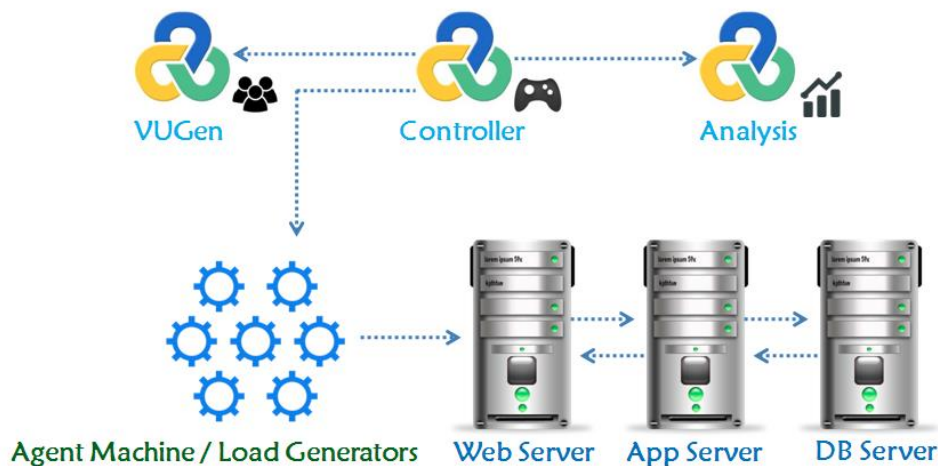
1. Which components have you used in Load Runner?

LoadRunner is a performance testing tool that includes following components:

1. **Virtual User / Vuser scripts:** It will allow you to record and replay script. And allow you to enhance your script
2. **Controller:** It will allow you to design the scenarios and execute the test with multiple users
 - Load Generators: It generates the load against the application
 - Agent Process: It will establish the connection between Load Generator & Controller
3. **Analysis:** It will allow to analyze the statistics and results and finding out the bottlenecks

Such components are used to simulate user activity, measure performance, and analyze results

HP LoadRunner Architecture



2. How can you set the number of Vusers in Load Runner?

In Load Runner, we can set the number of Vusers in the following ways:

- **Controller:** In the Controller, go to "Scenario" > "Vusers" and set the "Number of Vusers" field
- **Runtime Settings:** In the Vuser script, go to "Runtime Settings" > "General" and set the "Number of Vusers" field
- **Command Line:** Use the command line option "-n" followed by the number of Vusers, for example: "lr -n 10"

Note: The number of Vusers can also be controlled through the Load Runner API and through external automation tools

3. What is Correlation?

Correlation is the process of capturing and replacing dynamic values in a script with unique values for each virtual user (Vuser). This ensures that each Vuser interacts with the application independently, simulating real-user behavior

- Handle session IDs, tokens, and other unique identifiers
- Simulate unique user inputs, like usernames and passwords
- Capture and reuse dynamic data, such as timestamps or order numbers

By correlating dynamic values, you ensure that your load test accurately reflects real-world usage and avoids errors caused by duplicate or hardcoded values

4. What is the process for developing a Vuser Script?

The process for developing a Vuser script in LoadRunner is:

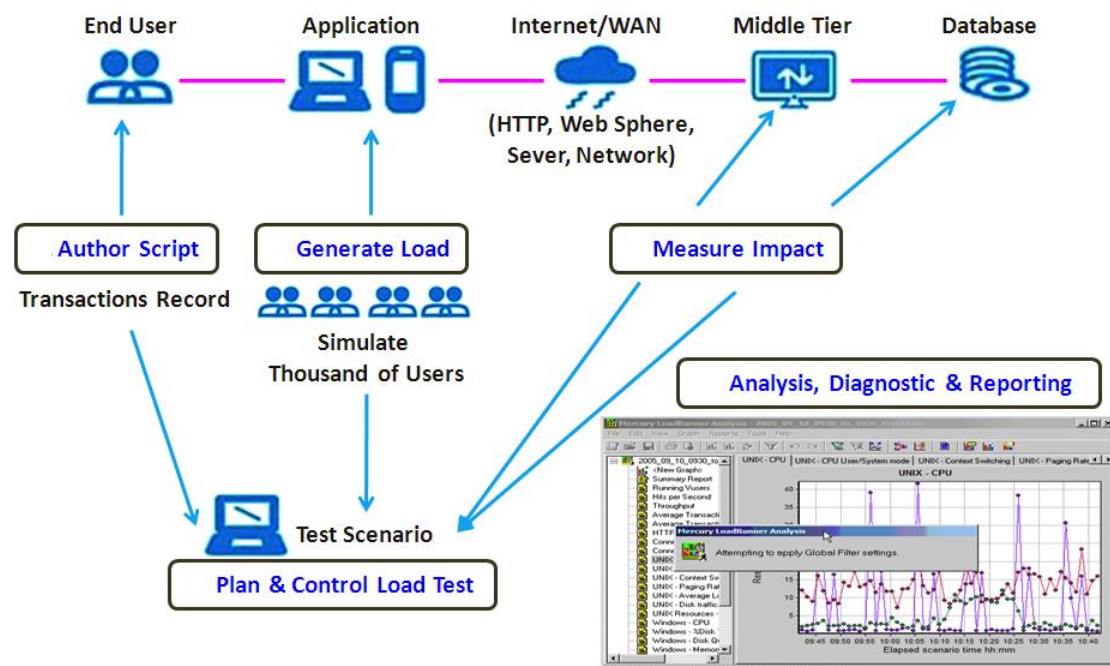
- **Record:** Capture user interactions with the application using LoadRunner's recording tool
- **Analyze:** Review the recorded script, identify dynamic values, and determine what needs to be correlated
- **Parameterize:** Replace dynamic values with parameters to enable unique data for each Vuser
- **Correlate:** Capture and replace dynamic values with correlated data
- **Script Enhancement:** Add think time, loops, conditional statements, and other logic to simulate realistic user behavior
- **Verification:** Validate the script's functionality and accuracy
- **Debug:** Test and debug the script to ensure it runs smoothly
- **Finalize:** Prepare the script for load testing by setting runtime settings and configuring logging

This process helps to create a robust Vuser script that accurately simulates real-user interactions with the application

5. How Load Runner interacts with the application?

Load Runner uses software agents called "protocols" to communicate with the application. These protocols mimic real-user interactions by sending requests to the application's servers and receiving responses. The protocols used depend on the application's architecture and technology stack
Here's how it works:

1. **application Selection:** LoadRunner selects the appropriate protocol (e.g., HTTP, Web Services, Citrix) based on the application's technology
2. **Script Recording:** LoadRunner records user interactions with the application using the selected protocol, capturing requests and responses
3. **Script Replay:** During load testing, LoadRunner replays the recorded script, sending requests to the application's servers using the same protocol
4. **Request/Response:** The application processes the requests and sends responses back to LoadRunner, which analyzes the responses to verify functionality and performance
5. **Data Correlation:** LoadRunner correlates dynamic data, such as session IDs or tokens, to ensure each virtual user interacts with the application independently
6. **Generate Load:** LoadRunner generates a large volume of virtual users, each executing the script and interacting with the application, simulating real-world usage



6. How many VUsers are required for load testing?

The main purpose of VUsers is to simulate the live environment. Universal formula to calculate the arriving rate to the system is Little's Law

$$N = Z * (R + T)$$

where N – number of VUsers

Z – Transactions per Second (TPS)

R – Response Time in seconds

T – Think Time in seconds

If you get the following data from the stakeholders i.e. TPS, Response Time and Think Time, number of VUsers can be calculated

e.g., TPS is 100, R is 3 sec and T is 2 sec then N will be

$$\begin{aligned} N &= 100 * (3+2) \\ &= 100 * 5 \\ &= 500 \end{aligned}$$

Peak load will be 500 VUsers

The number of VUsers (Virtual Users) required for load testing depends on several factors, including:

- 1. Expected user load:** Estimate the number of real users who will be using the application simultaneously
- 2. Application complexity:** More complex applications may require more VUsers to simulate realistic usage
- 3. Test goals:** Identify what you want to achieve with load testing (e.g., peak performance, stress testing)
- 4. Hardware resources:** Ensure the load testing environment can handle the desired number of VUsers.

Here are some general guidelines:

- **Low-load testing:** 10-50 VUsers (e.g., small applications, development testing)
- **Medium-load testing (Stress):** 50-200 VUsers (e.g., medium-sized applications, performance testing)
- **High-load testing (Scalability):** 200-1,000 VUsers (e.g., large applications, stress testing)
- **Extreme-load testing (flood):** 1,000+ VUsers (e.g., very large applications, extreme stress testing)

Here, the key is to simulate realistic user behavior and gradually increase the load to measure the application's performance and identify bottlenecks

7. What is the relationship between Response Time and Throughput?

- **Response Time:** The time taken by an application to respond to a user request or action, i.e. how long a user must wait for a response
- **Throughput:** The number of user requests or actions an application can handle within a given time period e.g., transactions (or any action) per second

The relationship between Response Time and Throughput:

- Inverse correlation: As Throughput increases (more requests handled), Response Time typically decreases (faster responses)
- Trade-off: Improving one metric can impact the other For example, optimizing for faster Response Times might reduce Throughput, and vice versa
- Balance: Aim for a balance between Response Time and Throughput to ensure a good user experience and efficient system performance
- In load testing, analyzing both Response Time and Throughput helps identify performance bottlenecks and optimize application performance

8. Advantages of Bugzilla

Bugzilla is a popular choice for bug tracking and management in software development teams

Advantages of bugzilla are given below,

- **Open-source:** Free to use and modify
- **Customizable:** Tailor it to your team's needs
- **Scalable:** Handles large numbers of bugs and users
- **User-friendly:** Intuitive interface for easy bug tracking
- **Searchable:** Powerful search functionality for quick bug lookup
- **Reporting:** Generate detailed reports on bug trends and metrics
- **Collaboration:** Assign, track, and discuss bugs with team members
- **Version control:** Track bugs across different product versions
- **Security:** Granular access control and security features
- **Community support:** Active community and extensive documentation
- **Integration:** Integrates with various tools and platforms
- **Stability:** Reliable and stable bug tracking system