

Module - 4 Assignment

Automation Core Testing (Load Runner Up)

- **Which components have you used in Load Runner?**

- In LoadRunner, various components are utilized to create and execute performance tests. These components work together to simulate user activity, monitor system behavior, and analyze performance metrics. Some of the key components include:
 - **Virtual User Generator (VuGen):** VuGen is used to record user interactions with an application and generate scripts that simulate these interactions. It supports multiple scripting languages like C, Java, and JavaScript.
 - **Controller:** The Controller component is used to design and execute performance tests. It allows you to define scenarios, allocate virtual users, set up load distribution, and monitor test execution in real-time.
 - **Load Generators:** Load Generators are machines responsible for simulating virtual users and generating load on the system under test. These machines execute the scripts created in VuGen under the control of the Controller.
 - **Analysis:** The Analysis component is used to analyze the results of performance tests. It provides various graphs and reports to identify performance bottlenecks, analyze system behavior under load, and measure key performance indicators.
 - **Monitoring:** LoadRunner integrates with various monitoring tools and protocols to gather performance metrics from servers, application components, and network devices during test execution. This data is used for analysis and troubleshooting.
 - **LoadRunner Agents:** Agents are used to collect performance data from servers and other monitoring points during the test execution. They work in conjunction with Load Generators to monitor system resources and gather performance metrics.
- These components together form the core infrastructure of LoadRunner, enabling users to conduct comprehensive performance testing of their applications and systems.

- **How can you set the number of Vusers in Load Runner?**

- In LoadRunner, the number of virtual users (Vusers) can be set using the Controller component.
- Here's how you can do it:
 - **Open Controller:** Launch the LoadRunner Controller application.
 - **Create a New Scenario or Open an Existing One:** You can either create a new scenario or open an existing scenario where you want to set the number of Vusers.
Define Vuser Groups: In the scenario, you can define one or more Vuser groups. Each Vuser group represents a set of virtual users that will execute a specific set of actions or transactions.
 - **Set the Number of Vusers for Each Group:** For each Vuser group, you can specify the number of virtual users you want to simulate. This can be done by entering the desired number directly into the appropriate field or by using the slider to adjust the number.
 - **Configure Load Distribution:** Optionally, you can configure the distribution of Vusers across different load generators if you have multiple load generators configured in your scenario.
 - **Validate and Run the Scenario:** Once you have set the desired number of Vusers for each group, validate the scenario to ensure it is configured correctly. Then, you can start the scenario to begin executing the performance test with the specified number of Vusers.

- **Monitor the Test Execution:** During the test execution, you can monitor the performance metrics and other relevant data to analyze the behavior of the system under the specified load.
 - By following these steps, you can easily set the number of Vusers in LoadRunner and conduct performance tests to evaluate the performance of your applications and systems under different loads.
- **What is Correlation?**
 - Correlation in the context of LoadRunner (or generally in performance testing) refers to the process of identifying and handling dynamic data within web applications.
 - When you record a script using a tool like LoadRunner, it captures the interactions between a user and the application.
 - However, many web applications use dynamic data, such as session IDs, timestamps, authentication tokens, etc., which are unique for each session or user interaction. These dynamic values are crucial for maintaining the state of the application and ensuring security.
 - During script replay, if these dynamic values are not handled properly, the script will likely fail because it will be sending incorrect or outdated data to the server.
 - Correlation helps in dynamically capturing and replacing these values during script replay, ensuring that the script behaves correctly and simulates real user interactions accurately.
 - The process of correlation involves:
 - **Identifying Dynamic Values:** Analyzing the recorded script to identify parameters or values that change dynamically between user sessions or interactions.
 - **Capturing Correlation Parameters:** Using LoadRunner's correlation mechanisms (such as automatic correlation or manual correlation) to capture and extract dynamic values from server responses during script replay.
 - **Parameterization:** Replacing the captured dynamic values with parameterized values that can be dynamically generated or extracted during script execution. This ensures that each virtual user gets a unique set of values as required by the application.
 - **Verification:** Verifying that the correlation process is successful by replaying the script and monitoring the requests and responses to ensure that the dynamic values are being correctly replaced.
 - By properly correlating dynamic values in scripts, you can create more realistic and reliable performance tests that accurately simulate user behavior and interactions with web applications.
 - **What is the process for developing a Vuser Script?**
 - Developing a Vuser script in LoadRunner involves several steps to accurately record user interactions with the application and create a script that simulates these interactions during performance testing. Here's a general process for developing a Vuser script:
 - **Planning:** Understand the objectives of your performance testing, including the scenarios you want to simulate, the user interactions to be recorded, and the key performance metrics to be measured.
 - **Recording:** Use LoadRunner's Virtual User Generator (VuGen) to record the interactions of a real user with the application. During recording, VuGen captures the HTTP/HTTPS requests and responses exchanged between the client and server.
 - **Enhancing the Script:** After recording, enhance the script by performing tasks such as parameterization, correlation, error handling, and think time insertion.

- **Parameterization:** Identify dynamic values (e.g., session IDs, user inputs) and replace them with parameters so that each virtual user gets unique values during script execution.
- **Correlation:** Identify and handle dynamic values that need to be correlated to ensure script replayability and accuracy.
- **Error Handling:** Implement error-handling logic to handle unexpected errors or responses from the server.
- **Think Time Insertion:** Insert think time into the script to simulate the time a real user would take between interactions.
- **Script Validation:** Validate the script by replaying it in VuGen to ensure that it accurately simulates user interactions and handles dynamic values correctly.
- **Script Debugging:** Debug the script to identify and resolve any issues or errors encountered during validation. Use VuGen's debugging tools, such as breakpoints, logs, and step-by-step execution, to troubleshoot the script.
- **Parameterization and Scenario Design:** Define parameters and scenarios in LoadRunner's Controller to customize the test execution, including the number of virtual users, ramp-up and ramp-down periods, and load distribution.
- **Test Execution:** Execute the performance test using LoadRunner's Controller. Monitor the test execution in real-time and collect performance metrics to analyze the behavior of the system under load.
- **Results Analysis:** Analyze the test results using LoadRunner's Analysis tool to identify performance bottlenecks, response times, throughput, and other key performance indicators. Use the insights gained to optimize the application's performance.
- **Iterative Improvement:** Based on the analysis of test results, iterate on the script, parameterization, and test scenarios to refine and improve the performance testing process.
- By following this process, you can develop effective Vuser scripts in LoadRunner to accurately simulate user interactions and evaluate the performance of your applications under various load conditions.

● **How Load Runner interacts with the application?**

- LoadRunner interacts with the application under test through the Virtual User Generator (VuGen), which records the interactions between a user and the application.
- Here's how LoadRunner interacts with the application:
- **Recording:** LoadRunner captures the interactions between the client and the server while a user navigates through the application. During recording, VuGen acts as a proxy between the client (e.g., web browser) and the server, intercepting and recording the HTTP/HTTPS requests and responses exchanged between them.
- **Protocol Support:** LoadRunner supports various protocols such as HTTP, HTTPS, Web Services, Java Messaging Service (JMS), Database (JDBC), Citrix, SAP, etc. Depending on the protocol used by the application, LoadRunner captures the relevant network traffic and protocol-specific messages during recording.
- **Parameterization and Correlation:** LoadRunner identifies dynamic values within the recorded script, such as session IDs, timestamps, or user inputs, and replaces them with parameters to ensure that each virtual user receives unique values during script execution. LoadRunner also performs correlation to handle dynamic values that need to be dynamically extracted from server responses.
- **Playback:** During script execution, LoadRunner replays the recorded script by sending the captured requests to the server and receiving the corresponding responses. The replay process simulates the user interactions with the application under test.

- **Think Time Simulation:** LoadRunner inserts think time between user actions to simulate the time a real user would take to interact with the application. Think time helps in creating a realistic simulation of user behavior and workload.
 - **Monitoring and Analysis:** LoadRunner monitors the performance of the application under test during script execution by collecting various performance metrics such as response times, throughput, server resource utilization, etc. These metrics are analyzed to identify performance bottlenecks, scalability issues, and areas for optimization.
 - **Load Distribution:** LoadRunner distributes the simulated load across multiple load generators to generate the desired level of concurrency and stress on the application. This allows testers to evaluate the performance of the application under realistic load conditions.
 - Overall, LoadRunner interacts with the application by recording user interactions, simulating these interactions during script execution, monitoring performance metrics, and analyzing the behavior of the application under various load conditions.
 - This comprehensive approach enables testers to conduct effective performance testing and identify performance issues early in the development lifecycle.
-
- **How many VUsers are required for load testing?**
 - The number of Virtual Users (VUsers) required for load testing depends on various factors including:
 - **Application Complexity:** Complex applications with multiple functionalities and components may require more VUsers to simulate realistic user traffic.
 - **Expected User Load:** The anticipated number of concurrent users accessing the application during peak usage hours influences the number of VUsers needed for load testing.
 - **Business Requirements:** Specific business requirements and service level agreements (SLAs) may dictate the minimum performance thresholds that need to be tested under load. The number of VUsers should be sufficient to meet these requirements.
 - **System Capacity:** The capacity of the application's infrastructure, including servers, databases, and network bandwidth, also plays a role in determining the required number of VUsers. Load testing should aim to stress the system to its capacity limits while maintaining acceptable performance.
 - **Scalability Testing Goals:** If scalability testing is part of the load testing objectives, the number of VUsers should be gradually increased to assess how the application scales with increasing load.
 - **Test Environment Constraints:** Constraints such as hardware resources, licensing limitations, and time constraints may impact the number of VUsers that can be used in the load test.
 - To determine the appropriate number of VUsers for load testing, it's essential to analyze these factors carefully and conduct performance modeling or pilot tests to estimate the optimal workload that accurately represents the expected user behavior and system load.
 - Additionally, iterative testing and tuning may be necessary to refine the number of VUsers and ensure that the load test effectively evaluates the performance of the application under realistic conditions.

- **What is the relationship between Response Time and Throughput?**
 - Response time and throughput are two key performance metrics used in load testing to evaluate the performance of an application. While they measure different aspects of system performance, there is a relationship between them.
 - **Response Time:**
 - Response time is the time taken for the server to process a user request and provide a response back to the client.
 - It includes the time spent waiting in queues, processing on the server side, and network latency.
 - Response time is typically measured in milliseconds or seconds.
 - Lower response times indicate better performance and user experience, as users experience faster interactions with the application.
 - **Throughput:**
 - Throughput is the rate at which the system is able to process incoming requests within a given time period.
 - It represents the number of transactions completed per unit of time (e.g., requests per second).
 - Higher throughput indicates better system capacity to handle a larger number of concurrent users or transactions.
 - Throughput is influenced by factors such as server capacity, network bandwidth, and application architecture.
 - The relationship between response time and throughput can be understood in the context of system performance and capacity:
 - **Inverse Relationship:** Generally, there is an inverse relationship between response time and throughput. As throughput increases (i.e., more requests are processed per unit of time), response time tends to increase. Conversely, as throughput decreases, response time tends to decrease.
 - **Saturation Point:** At a certain point, increasing the load on the system beyond its capacity leads to a decrease in throughput and an increase in response time. This point is often referred to as the saturation point or breaking point, where the system becomes overloaded and performance degrades significantly.
 - **Optimization:** Performance optimization efforts aim to find a balance between response time and throughput. By optimizing system resources, such as server configurations, database queries, and network infrastructure, it's possible to improve both response time and throughput simultaneously, maximizing the system's performance under load.
- **To test the Performance testing on “Tops Technologies website” :-**
 - <https://www.saucedemo.com/>
 - 1. to Record all top level menu
 - 2. to Record minimum 10 Vuser on this website
 - 3. save all (Script,Design,Graph) create a normal script of above website with correlate using hp default website.



SAUCEDEMO.side