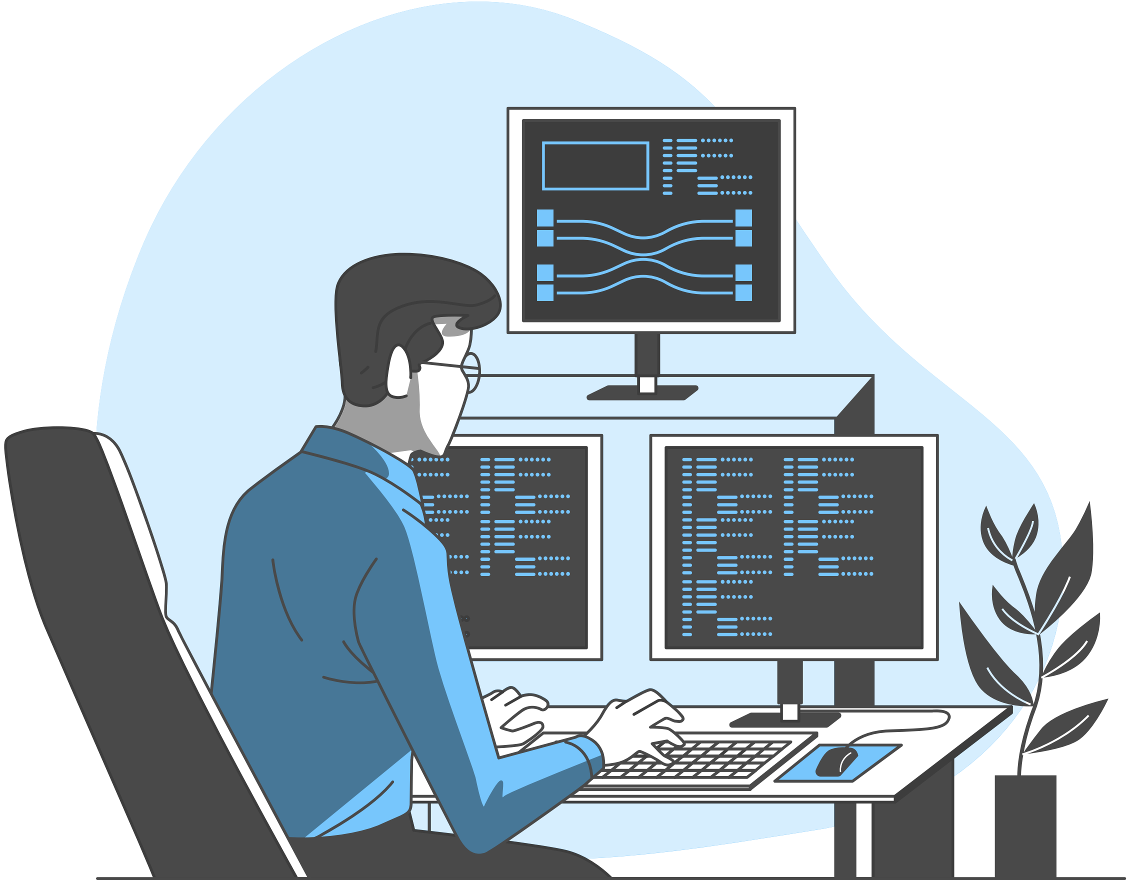
CDIP
Center for Development of IT Professionals



Professional Software Testing & Quality Assurance

Instructor

Parvez Hossain

JMeter

JMeter is an open-source performance testing tool developed by the Apache Software Foundation. It is designed to test the performance, load, and functional behaviour of various types of software applications. **JMeter allows testers to simulate different types of users, requests, and scenarios to evaluate how an application performs under different levels of load.**

Key features of JMeter include:

- **Load Testing:** JMeter can simulate a large number of virtual users (threads) making requests to a target application simultaneously. This helps in assessing the application's responsiveness and stability under heavy loads.
- **Stress Testing:** By gradually increasing the load on an application, JMeter helps identify its breaking points and performance bottlenecks under extreme conditions.
- **Performance Metrics:** JMeter provides various performance metrics like response time, throughput, latency, and error rates, helping testers analyze the application's behaviour and performance.
- **Test Plans:** Test plans in JMeter consist of thread groups, samplers (HTTP requests, database queries, etc.), and other components that define the testing scenarios. Testers can configure thread counts, ramp-up periods, and loop counts to simulate different user behaviours.
- **Reporting:** JMeter generates detailed reports and graphs based on test results, which helps in identifying performance bottlenecks and making informed decisions about application improvements.

JMeter Test Plan

Creating a test plan hierarchy in JMeter is essential for organizing and managing your performance tests effectively.

- **Test Plan:** Start by creating a Test Plan element, which serves as the root of your test structure.
- **Thread Group:**

→ Add one or more Thread Groups under the Test Plan. Thread Groups simulate the concurrent user load on your application. You can have different Thread Groups for various test scenarios (e.g., different user behaviours or volumes).

→ Each Thread Group can have its own configuration settings for the number of threads (users), ramp-up period, and loop count.

- **Controllers:** Within each Thread Group, you can use controllers to define the flow of your test. Common controllers include:

→ **Samplers** (HTTP Request, JDBC Request, FTP Request, etc.): These are the actual requests or actions your virtual users will perform.

→ **Logic Controllers** (e.g., If Controller, Loop Controller, Transaction Controller): These control the flow and execution of requests.

JMeter Test Plan

- **HTTP Requests:**

  → Under the controllers, you can add HTTP Request samplers to simulate HTTP traffic to your web application. Configure these samplers with the necessary parameters such as server name, path, method, and any required headers or parameters.

- **Listeners:** Listeners are essential for collecting and analyzing test results. Add listeners at different levels of your hierarchy to capture data.

  → **Thread Group Listeners**: Collects data at the Thread Group level.

  → **Sampler Listeners**: Collects data at the sampler level.

  → **Test Plan Listeners**: Collects data at the Test Plan level.

- **Configuration Elements:** Use Configuration Elements to set up variables, user-defined variables, CSV Data Set Config, or other test-wide settings.
- **Assertions:** Assertions help you verify that your application is responding correctly. Add them to your samplers to validate responses.
- **Pre-Processors and Post-Processors:** Use Pre-Processors to modify requests before they are sent and Post-Processors to extract or manipulate data from responses.

Common Interview Questions

- **What is load testing, and why is it important?**

  → Load testing is the process of testing an application under expected load conditions to ensure it performs well. It helps identify performance bottlenecks and weaknesses in the system.

- **Explain the typical components of a JMeter test plan.**

  → A JMeter test plan consists of Thread Groups, Samplers, Listeners, Controllers, Timers, and Config Elements.

- **What is a Thread Group in JMeter?**

  → A Thread Group represents a group of virtual users that will execute a test plan.

- **What is a Sampler in JMeter?**

  → Samplers are responsible for sending requests to the server. Examples include HTTP Request, JDBC Request, and FTP Request.

- **What is a Listener in JMeter?**

  → Listeners collect and display the results of test executions. Common listeners include View Results Tree, Summary Report, and Aggregate Report.

Common Interview Questions

- **What is a Controller in JMeter?**

  → Controllers are used to control the flow of requests and logic in a test plan. Examples include Simple Controller, Loop Controller, and If Controller.

- **How do you parameterize user data in JMeter tests?**

  → You can use variables and CSV data files to parameterize user data in JMeter tests.

- **Explain the use of Assertions in JMeter.**

  → Assertions are used to validate the correctness of server responses. They help determine whether a request was successful or not.

- **What is the purpose of the Thread Group configuration settings in JMeter?**

  → Thread Group settings specify the number of threads (virtual users) and the ramp-up period (time to start all threads) for a test.

- **How do you analyze and interpret the results generated by JMeter?**

  → JMeter provides various listeners and reports to analyze results. Key metrics to look at include response times, throughput, error rates, and resource utilization.

Common Interview Questions

- **What is the purpose of the Ramp-Up Period in JMeter?**

  → The Ramp-Up Period determines how quickly JMeter will start all the threads in a Thread Group. It's used to simulate a gradual increase in user load.

- **What is latency in JMeter load testing ?**

  → In JMeter load testing, latency refers to the amount of time it takes for a request to travel from the client (JMeter) to the server and back.

Client-Side Latency: This is the time it takes for a request to leave the JMeter client and reach the server. It includes factors like network latency (the time it takes for data to travel over the network) and any additional delays caused by JMeter itself.

Server-Side Latency: Once the request reaches the server, it measures the time it takes for the server to process the request and send a response back to the client. Server-side latency includes processing time, database queries, application logic execution, and any other server-specific delays.

Latency is typically measured in milliseconds (ms) and is an important metric because it directly impacts the user experience. High latency can result in slower response times, which can frustrate users and lead to a poor user experience.

CDIP
Center for Development of IT Professionals

Thanks!

Do you have any questions?

Email: parvez9605@gmail.com
WhatsApp: +88 01772880239