1. What is Load/Stress/Performance testing?

Load testing: is the process that simulates actual user load on any application or website. It checks how the application behaves during normal and high loads. This type of testing is applied when a development project nears to its completion.

Examples of performance testing are : Checking concurrent users, HTTP connections or checking suitable response time.

Stress testing: is a type of testing that determines the stability and robustness of the system. It is a non-functional testing technique. This testing technique uses an auto-generated simulation model that checks all the hypothetical scenarios.

Examples of Load testing are:Test of a word processor by making changes in the large volume of data.

Performance testing: is a type of testing for determining the speed of a computer, network or device. It checks the performance of the components of a system by passing different parameters in different load scenarios.

Example of Stress testing is casually shut down and restart ports of a large network.

2. Why do we do the Performance testing?

a. Experts believe that mobile application errors are much higher than what has been reported. Mobile applications struggle with network issues, especially when the server is congested. And if the applications are running on unreliable mobile networks, it becomes even more difficult. Some of the problems that apps face in such a situation are:

- Issues in downloading images or broken images.
- Giant black holes in content feeds
- Booking or checkout errors

b. Poor application experience means frustrated customers, which translates into lost revenues. Research shows that over 47% of the respondents, when faced with a broken image, would exit the application and transact on a different platform.

c. Application speed changes as per region. It is important to update an app country-wise and test it for the same. Internal testing should be done on the performance of the applications in various speeds and different networks. Some countries have 2G connection, some have 3G and others 4G. It is important to check whether users of the application across the world can use it conveniently, without any network issues.

3. Different tools through which we can do the performance testing.

(i) LoadRunner:

It is the most widely used tool to test applications, measure system behavior and their performance under varying loads. The tool is used to simulate thousands of concurrent users and record the variations in system performance. Afterward, leveraging those results, the performance of key components of the application can be analyzed. This tool supports all advanced technologies like Ajax, Flex, HTML5.0, Java, SOAP, Citrix along with all other legacy technologies.

Tool Functionality: It supports testing of a wide range of applications along with continuous testing. The tool effectively identifies performance bottlenecks and displays the root cause analytics.

(ii) Apache JMeter:

It is an important open-source tool used to test the performance of both static and dynamic applications. It is a Java-based application used for load and performance testing, which is used to test Web applications, SOAP & REST web services, FTP, databases and more. Interestingly, JMeter is also a widely used performance testing tool, which is effectively used to load test web and mobile applications and measure their performance.

Tool Functionality: Originally it was designed for testing web applications, but later on it has been expanded to test other test functions. It is also used for simulating a heavy load on a server or a group of servers and essentially checks application performance under varied load conditions.

(iii). WebLOAD:

This is a widely used tool by enterprises for testing web and mobile applications. It is an effective tool as it combines performance, scalability, and integrity as a single process for testing applications. It is an alternative to LoadRunner.

Tool Functionality: This tool is used to create mobile load tests either by recording native mobile apps or browser-based apps, directly from the mobile handset. The ease of the tool is; the user can edit any scripts, add parameters and perform validation checks. After creating mobile scripts, it executes the mobile load tests and this tool integrates with a real mobile device.

Q4. What are the environments?

An environment is a set of variables you can use in your Postman requests. You can use environments to group related sets of values together and manage access to shared Postman data if you are working as part of a team.

Q5. What is the workload model?

An inaccurate workload model can lead to misguided optimization efforts in your production system, delayed system deployment, outright failures, and an inability to meet service-level agreements (SLA) for the system. Having the right workload model is crucial for the reliable deployment of any system intended to support a large number of users in a production environment.

To achieve a viable WLM you need to:

- Proactively monitor user and system activities and performance in your production environment.
- Identify symptoms of failure, including longer-than-acceptable response times (a reasonable SLA per your system dictates), application errors and unhandled exceptions, and system crashes.
- Create an accurate simulation of all use cases for your system.

Activities involved in WLM (Workload Modeling)

- Test Objectives Identification.
- Application Understanding
- Key Scenarios Identification
- Determining Navigation Paths of Key Scenarios

6. What is the use of the workload model?

Importance of WLM (Workload Modeling)

Performance Scenarios Identification: The fundamental activity of the Workload model is to understand the application and identify its performance scenarios.

Performance Test SLAs: Performance testing teams translate AUT non-functional requirements into performance test SLAs through workload model.

Makes Communication Easier: Workload model makes it easy for the performance testing teams to communicate the AUT performance scenarios and users' distribution on them with all the application stakeholders.

Test Data Preparation: Workload model helps in identifying the type and amount of test data which is always required before the working on the tool is started.

Required Number of Load Injectors: You always require a lot of infrastructures to successfully conduct the performance testing activity. Incorrect results are produced if the application is tested with inadequate infrastructure. Normally users load is simulated from multiple machines (i.e. load injectors) for accurate testing which is also identified from the Workload model.

7. What is User’s journey?

User Journey Monitoring bridges the gap between web analytics and usability testing, showing the real-time impact of errors and slowdowns on user behavior and on the bottom line.

The Dynamic user Journey monitor approach gives users necessary context to the results from analytics and user testing. Where web analytics can tell you “what happened” and user testing can show you “how users react”, Dynamic user journey monitoring can provide insight into “why this is happening”, “where this is happening” and “how often it happens”.

8. What is Thread?

A thread is the smallest unit of work that a system can execute.

Thread testing, a software testing technique used during the early integration testing phase to verify the key functional capabilities that carry out specific tasks. These kinds of techniques are very helpful if an application is of type that uses client server architecture.

Performing Thread testing on valid business transactions through the integrated client, server and network is very critical. Threads are integrated and tested incrementally as subsystems and then performed as a whole system.

9. What are samplers ?

Samplers in JMeter are added as a child of Thread Groups. These are used to send different types of requests to the server. Once the sampler request is processed by the server, its response is returned to JMeter and the same can be viewed and analyzed in terms of different performance parameters like response time, Hits per second, throughput, etc.

We can launch the Samplers in JMeter by following the path

Right Click on Thread Group -> Hover over Add -> Hover Over Sampler -> Click on the required Sampler

Some of the most widely used Sampler in JMeter is-

- HTTP Request – Used to send HTTP/HTTPS requests to the server. This is the most widely used sampler for testing Web-based applications.
- JDBC Request – Used to send SQL queries to a database server.
- SOAP/XML-RPC Request – Used to send SOAP requests to a SOAP web service.
- Test Action – This is a special type of Sampler, which doesn’t send a request to the server instead it is used to introduce pauses in a test.

- FTP Request – Used to send file puts and get requests to an FTP server.

10. What is a listener?

Listeners give the results of our test plan execution. There are various ways in which Jmeter provides results like a tree, graph, table, simple text, etc. By collecting and analyzing these values we can filter out the necessary values and share the report with the client. We can save the results from any listener into the file that is in our desired path by browsing the file and the result can be saved with .csv or. XML or .jtl extensions.These listeners can be added directly under the test plan or under any thread group.

Various Listeners in JMeter

- View Results in Table
- Aggregate Graph
- Aggregate Report
- Aggregate Report
- Assertion Results
- Adding Listeners to Test Plan

11. Why do we use Request Defaults?

HTTP Request Defaults is a very basic and key element of JMeter. This config element is used when all requests in the JMeter script are sent to the same server. You can add a single HTTP Request Defaults element under Test Plan with the proper server name or IP address in the field 'Server Name or IP'. Now, when you add HTTP Request controllers, then leave the 'Server Name or IP' field empty. The controllers will inherit this field value from the HTTP Request Defaults element. Make sure you have provided the correct method type in each HTTP Request.

Advantages:

- It makes JMeter script simple and easily maintainable. If server name or IP changes, then you do not need to change those details in each HTTP Request. Just make changes in HTTP Request Defaults and that will be applicable for all the HTTP Requests.
- It reduces the chances of missing the URL/port changes from HTTP Requests
- It also avoids data duplication

HTTP Request Defaults has the following input fields:

Name: To provide element name

Comments: To provide arbitrary comments (if any)

Protocol [http:]: Either http or https (Do not use ://)

Server Name or IP**:** Domain name or IP address of the web server without including prefix http://

Port Number: Port number of Web-server. In case there is no port number then keep it blank.

Path: It is a path to the resource. You can keep it blank for home page request or use /.

Content Encoding: The encoding to be used for the request

Connect: Number of milliseconds to wait for a connection to open.

Response: Number of milliseconds to wait for a response