

PERFORMANCE TESTING

Performance testing is the practice of evaluating how a system performs in terms of responsiveness and stability under a particular workload. Performance tests are typically executed to examine speed, robustness, reliability, and application size.

LOAD TESTING

Load testing is the process of putting simulated demand on software, an application or website in a way that tests or demonstrates its behavior under various conditions.

Introduction of Load Testing

In software testing, load testing is an integral part of performance testing under [non-functional testing](#).

Load testing is testing where we check an application's performance by applying some load, which is either less than or equal to the desired load.

Here, load means that when **N-number** of users using the application simultaneously or sending the request to the server at a time.

Load testing will help to detect the maximum operating capacity of an application and any **blockages** or bottlenecks.

It governs how the software application performs while being accessed by several users at the same time.

The load testing is mainly used to test the **Client/Server's performance and applications that are web-based**.

In other words, we can say the **load testing** is used to find whether the organization used for comparing the application is necessary or not, and the

performance of the application is maintained when it is at the maximum of its user load.

Generally, load testing is used to signify how many concurrent users handle the application and the application's scale in terms of **hardware, network capacity** etc.

The Objective of Load Testing

The main objective of using the load testing is to fulfill the following aspects:

- The load testing is used to perform the **maximum quantity** of software applications without important performance breakdown.
- It is used to govern the scalability of the application and allows various users to access it.
- It is used to identify the total count of users that can access the application simultaneously.
- The load testing is used to determine whether the latest infrastructure can run the software application or not and determine the sustainability of the application concerning extreme user load.

Why is load testing important?

The load testing is essential because of the following factor:

- It guarantees the system and its consistency and performance.
- The load testing is necessary if any code changes occur in the application that may affect the application's performance.
- It is important because it reproduces the real user scenarios.
- It helps find the system's bottlenecks.

RULES FOR LOAD TESTING

During the execution of the load testing, a test engineer should follow the below rules:

- A test engineer tries to evade downloading images on the site.
- Once the application becomes functionally stable, load testing should be planned.
- The reliability of response time concludes the past period should be logged and the same should be compared with several test runs.
- For each scenario or script number of users should be decided.

Load Testing Process

The Load testing process will be completed in the following steps:



Step1: Test environment setup

- In the first step, we will set up the test environment to execute the load testing to ensure the testing can be done appropriately.
- And the test environment should be set up near to the production environment as likely in terms of **network, hardware, software specifications etc.**

Step2: Load the test scenario or specify the performance criteria

- In the next step, we will define the performance criteria, which contain the response time, reasonable limits on throughput, and the load test transaction.

- And then, we create the load test scenarios, which ensure the success criteria are finalized.
- In the load testing, transactions are decided for an application and data is set for each transaction.

Note: A test scenario is a combination of scripts, and virtual users executed during a testing session. The Scenarios could be of two types, either manual or goal oriented.

For example, In the LoadRunner testing tool, the scenarios are created with the **LoadRunner controller's** help.

STEP 3: EXECUTION OF TEST SCENARIOS

- Once we successfully create the load test scenarios, we will execute the particular test scenarios.
- But before we execute the load test scenarios, we have to set the different configurations and matrices to collect the information.
- The load on the server is matched by consecutively several virtual users to complete the tasks concurrently.
- And we can execute the entire scenario in virtual user groups or individual virtual users.

STEP 4: ANALYSIS OF THE TEST RESULT

- After executing the load test scenarios, we will analyze the test results.
- The load test Scenario can be inspected with the help of **LoadRunner** online monitors like:
 - **System resource**
 - **Run-time transaction**
 - **Network delay**
 - **Web resource**

STEP 5: RE-TEST

- The last step of the load testing process depends on the test result because if the test fails, we have to perform the same process repeatedly until the test result is passed and all the issues and bottlenecks are fixed.

EXAMPLES OF LOAD TESTING

Let see **some real-time example** where we can see the massive failure of the particular application as they did not perform the load testing:

Example 1

E-commerce websites capitalize deeply in advertising campaigns but not in Load Testing to guarantee ideal system performance, and they get massive users at the same time. Because of that, some very popular sites have suffered a serious failure

Example 2

Amazon lost \$66,000-\$66,240 per minute because the amazon.com server crashed heavily by users' traffic for 30 minutes in 2013.

And during a festival offer, an Airline website cannot handle 10000+ users at the same time.

Example 3

More people have a habit of booking a flight ticket throughout holidays or on the days when an air company has an offer.

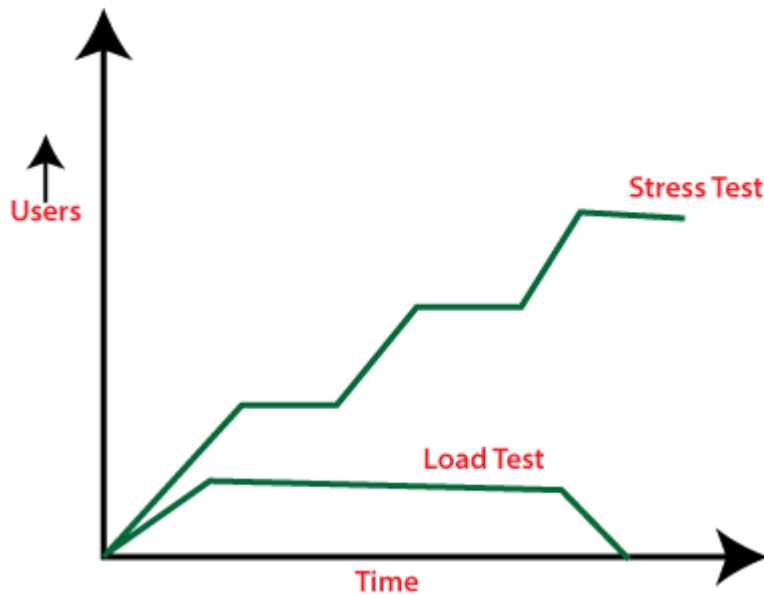
As we can see, more people incline to buy products during a promotional event like **Diwali, Black Friday, big billion days sales**.

And if a website or an application crashes in such an event, users may leave the website and go to a competitor's application, leading to loss of revenue and market share. Those scenarios may occur because we do not perform the load testing on the system.

Note: Based on a survey if a site is crashed or slow, 70-75% of the users would leave the website. If the website or an application did not load in 3 seconds, 40-50% of the users said they would buy elsewhere.

Difference between Load and Stress testing

The major difference between load and stress testing is listed in the following table:



Load Testing	Stress Testing
Using load testing, the test engineer can detect the bottleneck and tell the cause of bottlenecks before deployment to the production server.	By using Stress testing, the test engineer can check the system capacity when the number of users suddenly increases before the system failure or crashes.

TYPES OF LOAD TESTING TOOLS

We have the various type of load testing tools to execute the load testing, which is as follows:



Load Testing Tools [Open-Source]

- To perform the load testing, we can use the open-source load testing tools as we have various load testing tools available in the market for free of cost.
- If we have a limited budget for the particular project, we can use open-source tools. But not every time, as they may not be as advanced as the paid load testing tool.
- JMeter is the most frequently used open-source load testing tool.

Manual Load Testing

- One of the most effective load testing approaches is the manual process to perform the load testing. Still, we cannot rely on it as it does not produce repeatable outputs and assessable stress levels on an application.
- And if we perform the load testing manually, it requires a lot of workforces, which is quite expensive compared to paid tools.

Load Testing Tools [Enterprise-class]

- The paid load testing tools are used to support many protocols; therefore, it can implement various types of applications like **Streaming Media, ERP/CRM, etc.**
- **LoadRunner** is the most popular licensed load testing tool.

IN HOUSE DEVELOPED LOAD TESTING TOOLS

- The particular organization uses the in-house developed load testing tools approaches to build their tools to perform the load tests on their application for understanding the importance of load testing

LOAD TESTING TOOLS

We have various types of load testing tools available in the market, where some are **commercial tools** and open-source tools. Let see some of the most common load testing are as follows:

- **LoadNinja**
- **Apache JMeter**
- **NeoLoad**
- **HP Performance Tester**
- **WebLoad**
- **LoadView**

ADVANTAGES AND DISADVANTAGES OF LOAD TESTING

ADVANTAGES

Some of the vital **benefits** of performing the load testing are as follows:

- Load testing helps us to detect the bottlenecks and performance-related issues before production.
- The load testing enhances the scalability regarding network, software and database for the system or software application.
- The major advantage of performing the load testing is reducing the cost failures, which also helps us minimize the system interruption risks.
- Customer's satisfaction is enhanced while using the load testing.

DISADVANTAGES

Following are the **drawbacks** of load testing:

- Load testing can only be executed if we have enough knowledge of any programming language as well as testing tools.
- Usage of load testing tools can be an expensive process because pricing depends on the number of virtual users supported.

SUMMARY

- While executing an application's performance testing, the load plays a vital role and helps to learn the software's effectiveness and ability or an application.
- If we ignored the load testing, it might cause financial losses.
- It is specifying as a type of **software testing** which controls a system's performance under real-life load conditions.
- It expands the scalability, performance issues, and constancy of the application before production is available.
- Using the various load testing tools, we can ensure to deliver a quality product that cannot crash while maximum numbers of users using it simultaneously.

STRESS TESTING

Stress Testing, which is an important part of **Performance testing** and used to check the behavior of an application by applying a load greater than the desired load.

INTRODUCTION OF STRESS TESTING

In software testing, stress testing is an important part of performance testing under non-functional testing.

Stress Testing is testing used to check the accessibility and robustness of software beyond usual functional limits. It mainly considers for critical software but it can also be used for all types of software applications.

It is also known as *Endurance Testing*, *fatigue testing* or *Torture Testing*.

The stress testing includes the **testing beyond standard operational size**, repeatedly to a **breaking point**, to get the outputs.

It highlights the error handling and robustness under a heavy load instead of correct behavior under regular conditions.

In other words, we can say that **Stress testing** is used to verify the constancy and dependability of the system and also make sure that the system would not crash under disaster circumstances.

To analyse how the system works under extreme conditions, we perform **stress testing** outside the normal load.

THE OBJECTIVE OF STRESS TESTING

The main objective of using stress testing is to fulfill the following aspects:

- The primary purpose of executing the stress testing is **to confirm that the software does not crash in lacking computational resources like disk space, memory, and network request.**
- The implementation of stress testing certifies that the system fails and improves effortlessly, known as **the recoverability process.**

- We can use stress testing to discover hardware issues, data corruption issues.
- Stress testing will help us to identify the security weaknesses that might sneak-in throughout constant peak load.
- It helps determine the software application's data integrity throughout the extreme load, which implies that the data should be in a dependable state after a failure.

FEATURES OF STRESS TESTING

Following are the basic features of stress testing:

- Stress testing also makes sure that unpredicted failures do not cause security issues.
- It is used to analyze the system works under rare circumstances and the system's behavior after a failure.
- Stress testing is used to check the system has saved the data before crashing or not.
- Stress testing guarantees to display a suitable error message when the system is under stress.

WHY DO WE NEED TO PERFORM THE STRESS TESTING?

We need to perform stress testing if we encounter the following situations:

Whenever **e-commerce or online shopping sites** announce a sale during the festival may witness a spike in traffic. Or when an article is mention in a top newspaper, its knowledges an unexpected flow in traffic.

If we fail to assist this sudden traffic can result in loss of profits and status. So, in that case, we need to execute the **Stress Testing to integrate such irregular traffic spikes**.

Stress testing is also needed to be performed for the below scenarios:

- When the system is under stress, it should display a proper error message.
- To inspect whether the system works under bizarre circumstances.
- If the system is failed under risky situations could result in huge profits loss.
- By implementing Stress Testing, we will be prepared for extreme conditions.

EXAMPLE OF STRESS TESTING

Let's see some **real-time examples** where we can discover the usage of Stress Testing.

Example 1: E-commerce website/ application

Throughout the new product releases, sales, holidays, promotions, and festival offers, the e-commerce application tends to get many users in a very short time.

Example 2: News website at the time of some major/viral event

The news website will crash or slow down when a major event happens; **for example**, when Michael Jackson passed away, a maximum number of news websites are slow down, or some of them also crashed.

To overcome this situation, we need to perform stress testing on the particular application and be prepared to recover quickly in case of any crashes or failure.

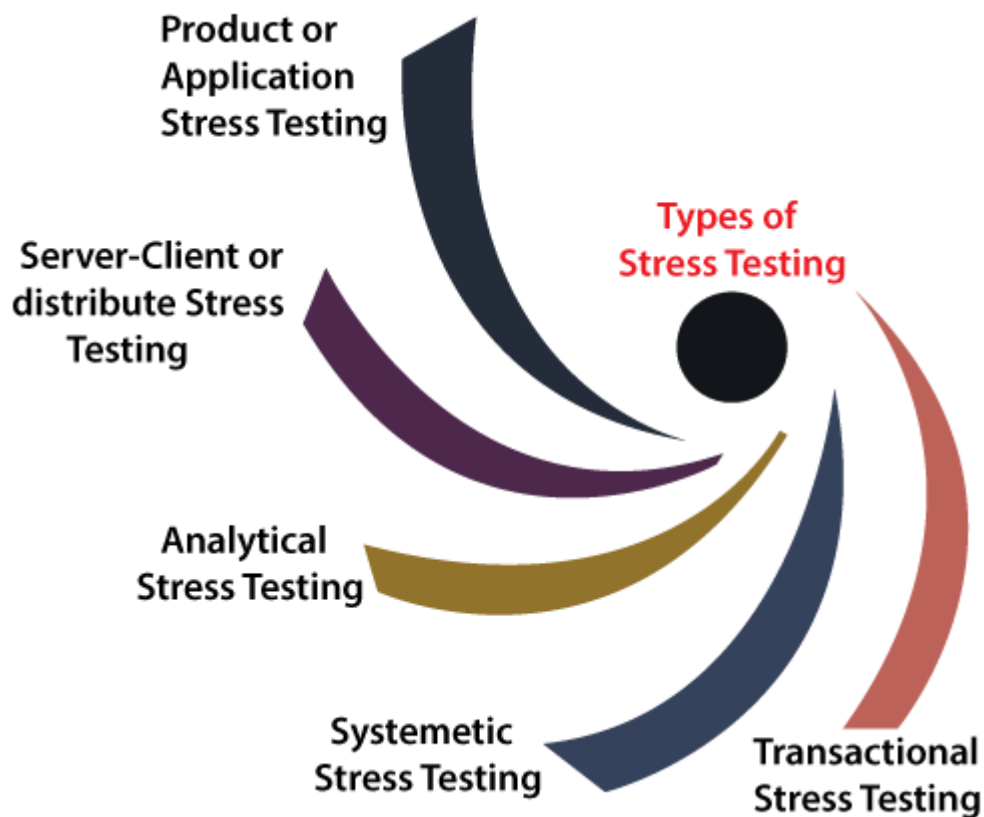
EXAMPLE 3: EDUCATION BOARD'S RESULT WEBSITE

Stress testing is important to perform on the education board's result website. On the day of some results, many students, users, and applicants will logins to the particular to check their grades.

Therefore, the execution of stress testing helps identify the failure of the application and evaluate the performance and recoverability when the ultimate load occurs in a short duration or when the result is out.

TYPES OF STRESS TESTING

Stress testing can be categories into various parts, which are as follows:



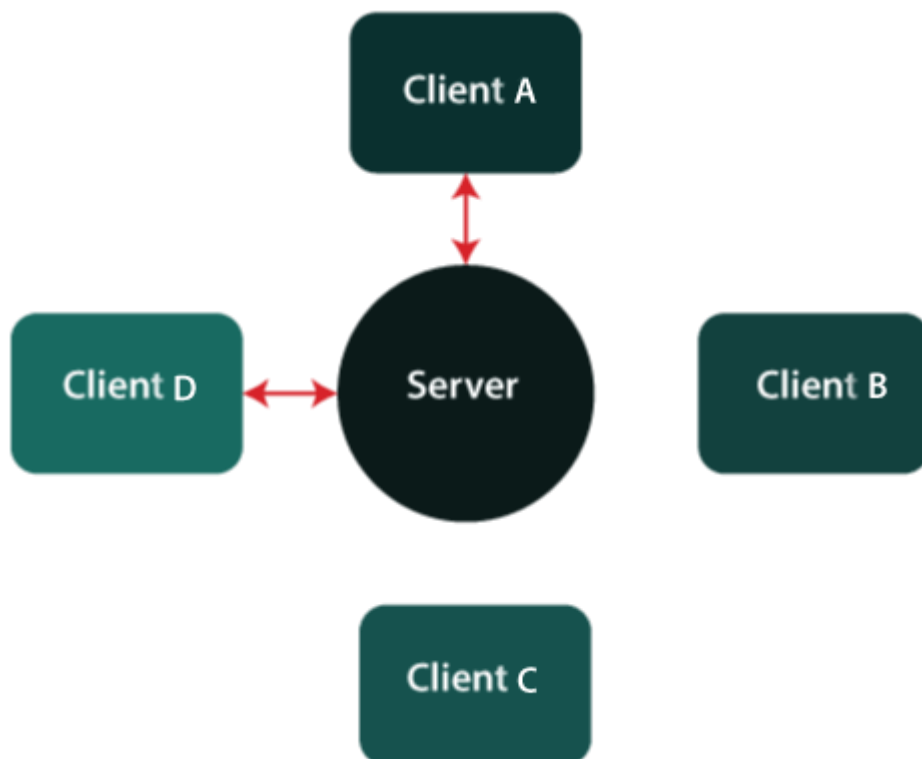
- **Product or Application stress testing**
- **Server-client or distribute Stress Testing**
- **Analytical Stress Testing**
- **Systematic Stress Testing**
- **Transactional Stress Testing**

Product or Application Stress Testing

- The application or product stress testing is mainly focused on determining the faults related to network issues, data locking, blocking, and a performance bottleneck in a software product.

Server-client or Distribute Stress Testing

- In this type of stress testing, all the clients related to the server are tested.
- The distribute stress testing is used to perform across all clients from the server.
- The server can communicate with **clients A and B**, but it cannot link with **clients C and D** when there is stress on the client-server system as we can see in the below image:



Analytical/ Exploratory Stress Testing

- Analytical or exploratory testing is used to execute the system with unusual constraints unlikely to occur in a real scenario.

- It is mainly used to identify the bugs in rare situations such as **a large number of users logged simultaneously or a database went offline when retrieved from a website.**

Let see some examples of **analytical Stress Testing** where such irregular conditions are used:

- When a large number of parallel users try to log into the application.
- Data is added in enormously large quantity in the database.
- When the website tries to reach it from the front end, and database linked to the website shuts down.

SYSTEMATIC STRESS TESTING

- It is combined testing used to execute the test across various systems running on a similar server.
- Using **systematic stress testing**, we can easily detect the bottleneck where data of one application blocks another application.

TRANSACTIONAL STRESS TESTING

- Another type of stress testing is **transactional stress testing**, which is used to implemented one or more transactions between various applications.
- The main objective of performing the transactional stress testing is **to enhance the system performance.**

PROCESS OF STRESS TESTING / HOW TO PERFORM STRESS TESTING

The stress testing process will be completed into the following steps:



Step1: Detect the testing environment

In the first step of stress testing, we will identify the network, software, and hardware configurations and tools available to achieve the stress test.

Step2: Find performance acceptance criteria

After identifying the testing environment, we will find the performance acceptance criteria, which help us categorize the metrics used to test the application's performance under stress.

And also, identifying the success criteria for a stress test, for example, the maximum load can apply to the application for it to fail.

Step3: Plan and design stress tests

In the next step of the stress testing process, we will plan and design a stress test plan, identify test scenarios etc.

Step4: Configure the test environment

Once the stress test plan has been created successfully, we will move to our next step where we create the test environment, tools and resources essential to perform each approach as features and components become available for test.

Step5: Implement test design

After the test environment's configuration, we will develop the stress tests resulting the test design best performs.

Step 6: Execute tests

In the next step, we will execute the particular test, observe and confirm the tests along with test data and output collection.

Step7: Analyze the results

In the last step of the stress testing process, we will analyze the outcomes, combine and share the respective teams' output data.

STRESS TESTING TOOLS

As we know that **stress testing** is part of **performance testing**, the tools used for performance testing can be used for stress testing. Therefore, we have various types of **Stress testing** tools available in the market, where some are commercial tools and open-source tools. Some of the most commonly **Stress testing** are listed below:

- **Apache JMeter**
- **NeoLoad**
- **Stress tester**
- **LoadRunner**

ADVANTAGES AND DISADVANTAGES OF STRESS TESTING

ADVANTAGES

Some of the vital **benefits** of performing Stress testing is as follows:

- Stress testing signifies the system's behavior after failure and makes sure that the system recovers quickly from the crashes.
- The most important advantage of executing the stress testing will make the system work in regular and irregular conditions in a suitable way.
- It determines the scalability and enhance the performance of the software.

DISADVANTAGES

Some of the most common **drawbacks** of Stress testing are as follows:

- Even in open-source tools like JMeter, a load testing environment is required, which should be as close to the production environment setup as possible.
- If we are writing the Stress test script, the person should have enough scripting knowledge of the language supported by the particular tool.
- If we are using stress testing, it will require additional resources, which makes this testing bit costlier.
- If we perform the **Stress Testing** manually, it became a tedious and complicated task to complete, and it may also not produce the expected results.

OVERVIEW

stress testing is used to assess the system under extreme situations. It can verify the system to recover back to normal status.

It is a type of **non-functional testing** and generally executed after the **functional testing**.

Stress testing entirely concentrate on testing the system under extreme load situations to detect its **breaking point** and see if suitable messages are shown when the system is not responsive.