|  |  |  |
| --- | --- | --- |
|  |  |  |
| **Performance Testing** | **Introduction** | **Introduction** |
|  |  | **Performance Testing Life Cycle** |

**Testing**

Software Testing can be Broadly classified into two types.

1. **Functional Testing (FT)**: To validate the functional behavior of the application, we will go for functional testing. It can be executed in 2 approaches.

🡪 Manual execution

🡪 Automation execution (selenium etc)

1. **Non-Functional Testing (NFT)** 🡪 Performance testing

🡪 Operational Readiness Testing (ORT)

🡪 Accesibility Testing

🡪 Security Testing

To Understand the basic Difference between these 2 types, Let us consider the following example

Let us say, you are trying to login to your face book account

Enter valid User Name and password and click the login button. You were able to login successfully. This means login functionality is working as expected.

Functional Test case 🡪 passed (positive test case)

Non- Functional Aspects of the Login Button.

* After you entered your login credentials and clicked on Login button, how long you had to wait to see your home page? Was it under 1 sec / 3 sec’s or 30 sec’s. (this in NFT Test case)
* What if 100 / 1000 people try to login at the same time? Will the login button response time is still under 1 sec?
* What happens if 1 million people are trying to login simultaneously, what happens now? Will our application able to handle this much load or it will crash?

PT can answer above Questions.

Now I hope you got the basic difference between Functional and Performance testing.

**Why Performance Testing is important and what problems it can solve**

These questions keep buzzing in our minds if any new software is going to release in market.

1. Will the end users be happy with the performance of our application after launch?
2. Will our IT systems handled the expected user load or will they crash?
3. Can our application process X number of orders per hour?
4. How much time does it take to process one order?
5. What is the max load that our application can handle before it can crash?
6. How does our application react to sudden spikes in the load?

**Performance testing can answer all these questions even before we launch our application.**

**Performance testing tools can simulate the workload expected on our Application using Virtual users.**

**Performance Testing:** To validate the behavior of an application under different work load (vusers) or against the expected user load.

1. Manual: do the performance testing by arranging all the Machines in a n/w and access the application concurrently by the real users.

Note: constraints are all are not the performing same action simultaneously, due to late in perform actions

1. Automation: do the performance testing of an application by simulating multiple users from single host or single machine

Objective:

The primary objecti ve of performance testing is to identify bottlenecks an application.

Bottleneck is a breakpoint where the application crashes or the environment down, Which can be called performance issue.

**Performance Test Life cycle**

**Requirement gathering**:

**Test Planing**

**Test Scripting**

-- **Test Execution and Monitoring**

**Analysis and Reporting**

**Requirement gathering:**

1. **V**user load
2. SLA – Response times, **Transactions per second**, Resource utilization (cpu, memory etc )
3. Tools to be used
4. **Flows to be tested \*\*\*\*\***
5. Application Access
6. Servers to be monitored

Web

Database

**Test Planing:**

**Test Scripting: prepare test scripts for all the business flows which are identified/got sign-off.**

**Test scripts should have proper naming conventions, asserions, correlations,**

**Parameteriztion.**

**Test Execution and Monitoring**

**Design and exeucute the multi user business scenario’s with identified testing types.**

Performance Testing type

Smoke test

Load test

**Stress Test**

Endurance tests -Memory leaks (unreleased memory which is no longer needed)

**Smoke test**: Test the behavior of an application at an early stages of performance test life cycle, to validate end to end performance test environment…gen. with min no.of users.

**Load test**: Test the behavior of an application by gradually increasing load on the server. First we go with

25% load then 50% and finally 100%.

**Stress test**: Test the behavior of an application by gradually increasing load on the server **Until unless Application crashes/ environment down.**

**Endurance test:** Test the behavior of the application in a prolonged duration (like 8h, 24 hours, I week etc)

The primary objective is to identify memory leaks in the application

**Memory leak**: un released memory which is no longer needed.

Gen in 1 hr test we may not find any memory leaks, i.e the reason we are doing for prolonged duration

**Capacity testing:** validate the behaviour of an application with different record length and different data formats

**Fail over testing**

Validate the recovery of load balancer, if any server fails in middle of execution.

**Monitoring :**

Monitor below metrices during load test execution to identify to identify performance bottleneck

1. **Transaction Response Time:**

**It is the time taken to execute end to end user action**. Let us consider an user clicks Login button.

C: 5:00:05 5:00:07 5:00:08

DB Server

Web or/and app server

client

.

5:00:15 5:00:12 5:00:10

Total Transaction Response time is 10 sec ( Exit time – Entry time)

When ever we see any delay in response time, we need to drill down to Web server or appserver or DB server or Network.

1. Hits/sec: it measeures the no.of requests made by the client to webserver.
2. Through put: it measures the amount of data transfer between client and webserver
3. Transactions per sec: it measures the no.of transactions to be executed per second

Eg: for The no.of transactions to be executed in a day is 500000.

In an hour

800000 🡪 800000/8 -> 100000

Per tra ….100000/60\*60

**Server side metrics**

**Web Server DBserver**

**.net IIS SQL Server**

**Java WebLogic Oracle**

**We need to check in 2 layers ----**

1. **OS Layer Or System Level: Windows, Unix/Linux etc**
2. **Server Level 🡪 Webserver 🡪 IIS, Weblogic etc**

**DB Server-🡪 SQL, Oracle etc**

**Analysis**: transaction response tme, hits/sec, throughput,TPS, **TFailcriteria**.

**VUser**: it is virtual user to simulate and execute instead of a real user.

**Workload or vuserLoad**: The no.of Users which are in execution

**User mix**: Distributing the workload across multiple scenario’s or scripts

Status codes

100 , 200 -- Client level status code

300 – Redirection level status code

400, 500 --- Server level status code

100 – continue ( informational , whether the request received )

2\*\* -- Success , action requested by the client was received , understand , accepted and processed successfully.

200 – Ok / Success (

300 – redirectional

4\*\* Bad reques -🡪 the request could not be understood by the server due to malformed syntax. The client should not repeat the request with out modification

401 - un authorized

402 – payment required

403 – forbiddern

404 – not found

405 - method not found

Etc

5\*\* server error

500 – internal server error

501 – not implemented

502 – bad gateway

503 – service not available

504 -gateway timeout

505—Http version not supported

Gmail id