

# *Approach to Performance Testing*

---

*Whitepaper*

Author: Indusa InfoTech & Services Pvt Ltd.

**Performance testing is a measure of the performance characteristics of an application. The main objectives of a performance testing are to demonstrate the response time, transaction throughput and other performance attributes of a system meets the specifications. This paper describes about the Performance Testing goals, objectives, components, Prerequisites for performance testing, phases of performance testing and metrics captured during the Performance Testing.**

CONCEPTS .....	3
GOAL OF APPLICATION DIAGNOSTICS .....	3
Objectives .....	3
“End-User” objectives: .....	3
“System” objectives: .....	3
“Management” objectives: .....	4
Components.....	4
Web Performance Objectives .....	4
Web Performance Testing .....	4
Web Performance Measurement.....	4
Problem Resolution and Activity Results .....	4
Capacity Planning .....	4
Approach to Performance Testing.....	5
Pre-Requisites for Performance Testing .....	5
(a) Performance Comparison Attributes .....	5
(b) Stable system.....	5
(c) Realistic test environment .....	5
(d) Controlled test environment.....	5
(e) Performance testing toolkit .....	6
(f) Understanding of Architecture.....	6
PHASES OF PERFORMANCE TESTING .....	7
(a) DESIGN PHASE: .....	7
(i) Requirements Gathering.....	7
(ii) Test Strategy Design .....	8
(iii) Define Operational Profile.....	9
(iv) Define Workload .....	9
(b) Build PHASE:.....	10
(i) Test Environment Setup .....	10
(ii) Record Test Scripts.....	10
(iii) Create Test Scenarios.....	10
(c) Execute PHASE:.....	10
(i) Baseline Tests: .....	10
(ii) Performance Tests .....	10
(iii) Scalability Tests .....	10
(iv) Benchmark Tests.....	10
(v) Report Generation.....	10
(d) Analysis & Tune: .....	11
(i) Diagnose Bottlenecks .....	11
(ii) Configuration Tuning.....	11
(iii) Quantify Improvements .....	11
References: .....	12

## **CONCEPTS**

*Web Performance Testing is executed through testing campaigns for stressing the web site and back-end systems with the amount of load simulating the real conditions of the field or to evaluate if the site/application will support the expected load following special situations (e.g. an advertisement campaign). That allows you to guarantee system performances under that load and to identify and help in fixing possible issues.*

## **GOAL OF APPLICATION DIAGNOSTICS**

1. *Web Performance Measurement aims at analysis and fast characterization of system and user behaviour in order to give fast feedback on any issues.*
2. *Web Performance Testing and Measurement aim at providing a measure of the actual performances of a web application, and an evaluation of performances that the application could provide, following a change of load; identifying, moreover, possible bottlenecks and providing useful advice about how to fix problems (tuning of system parameters, modification of software or hardware upgrade).*

## **Objectives**

There are different objectives achievable with a web performance testing and measurement. End-users take advantage of a better application, system managers use that information to improve performances of their systems and, last but not least, management can obtain useful information about the business of their company.

### **“End-User” objectives:**

- To find average response time of pages and transactions, slowest and fastest pages
- To make sure main pages (e.g. home page) can be downloaded within acceptable time (e.g. 10 seconds)
- To find out maximum number of concurrent users, sessions and transactions that the application is able to support still providing a high level of service
- To find out maximum number of concurrent users, sessions and transactions that the application is able to support without system crash
- To characterize more frequent user paths, the most used starting and exiting page
- To identify main reasons of site abandonment.
- As the user base grows, the cost of failure becomes increasingly unbearable. To increase Confidence and to provide an advance warning of potential problems in case of load conditions, analysis must be done to forecast performance under load.

### **“System” objectives:**

- To correlate system resource utilization with load
- To find out possible actual hardware bottlenecks and prevent new ones (capacity planning)
- To tune all the web application components to support as much load as possible using actual hardware
- To find out how the application works when overloaded

### **“Management” objectives:**

- To provide an objective measure of the usage of the site (e.g. for an e-commerce site, it could be the number of electronic carts and the number of objects sold)
- To provide a “business view” of the previous data (e.g. how performance issues have affected the business)
- During a design/redesign of a module or a part of the system, more than one alternative presents itself. In such cases, the evaluation of a design alternative is the prime mover for an analysis.
- Modern applications are not entirely designed from the ground up. Use of third-party tools and components is almost a given. In such scenarios, the comparison of two or more systems must be undertaken.
- Capacity planning is usually done with a view to the future. The results are used as ipso facto for decisions regarding the purchase of additional hardware resources, software tools, etc

### **Components**

To achieve the objectives just described, a complete approach is needed. This approach should be based

on sound methodology, a set of tools and a high level of know-how. The Telecom Italia Lab approach is based on the following components:

**Web Performance Objectives** – to identify, according to the previous classification (end-user, system, Management), what kind of results you need. Actually it’s possible to address all three categories or just one or two

**Web Performance Testing** – to verify, in test plant, that the application is able to support the entire load expected and even more (e.g. the load could increase in an unpredictable way after an advertising campaign). For a more detailed analysis, this activity could be done both inside and outside proxy and firewall

**Web Performance Measurement** – to measure the actual performance of the application in the field. This could be done by identifying user behaviour, back-end system response time, maximum number of concurrent users (**Web Log Analysis**), monitoring, at the same time, hardware and software resources utilization (**Resource Monitoring**) and the end user experience (**End-to-end Monitoring**)

**Problem Resolution and Activity Results** – to support client by providing the source required to fix any bottlenecks

**Capacity Planning** – to ensure that adequate hardware and software resources will be available when needed in the future. Capacity planning activity is carried out using all information from other components

## **Approach to Performance Testing**

### ***Pre-Requisites for Performance Testing***

We can identify six pre-requisites for a performance test. Not all of these need be in place prior to planning or preparing the test (although this might be helpful), but rather, the list defines what is required before a test can be executed.

#### **(a) Performance Comparison Attributes**

As a foundation to all tests, performance requirements should be agreed prior to the test. This helps in determining whether or not the system meets the stated requirements. The following attributes will help to have a meaningful performance comparison.

- **Quantitative** - expressed in quantifiable terms such that when response times are measured, a sensible comparison can be derived.
- **Relevant** - a response time must be relevant to a business process.
- **Measurable** - a response time should be defined such that it can be measured using a tool or stopwatch and at reasonable cost.
- **Realistic** - response time requirements should be justifiable when compared with the durations of the activities within the business process the system supports.
- **Achievable** - response times should take some account of the cost of achieving them.

#### **(b) Stable system**

A test team attempting to construct a performance test of a system whose software is of poor quality is unlikely to be successful. If the software crashes regularly, it will probably not withstand the relatively minor stress of repeated use. Testers will not be able to record scripts in the first instance, or may not be able to execute a test for a reasonable length of time before the software, middleware or operating systems crash.

#### **(c) Realistic test environment**

The test environment should ideally be the production environment or a close simulation and be dedicated to the performance test team for the duration of the test. Often this is not possible. However, for the results of the test to be realistic, the test environment should be comparable to the actual production environment. Even with an environment which is somewhat different from the production environment, it should still be possible to interpret the results obtained using a model of the system to predict, with some confidence, the behavior of the target environment. A test environment which bears no similarity to the actual production environment may be useful for finding obscure errors in the code, but is, however, useless for a performance test.

#### **(d) Controlled test environment**

Performance testers require stability not only in the hardware and software in terms of its reliability and resilience, but also need changes in the environment or software under test to be minimized. Automated scripts are extremely sensitive to changes in the behavior of the software under test. Test scripts designed to drive client software GUIs are prone to fail immediately, if the interface is changed even slightly. Changes in the operating system environment or database are equally likely to disrupt test preparation as well as execution and should be strictly controlled.

### (e) Performance testing toolkit

The execution of a performance test must be, by its nature, completely automated. However, there are requirements for tools throughout the test process. Test tools are considered in more detail later, but the five main tool requirements for our 'Performance Testing Toolkit' are summarized here:

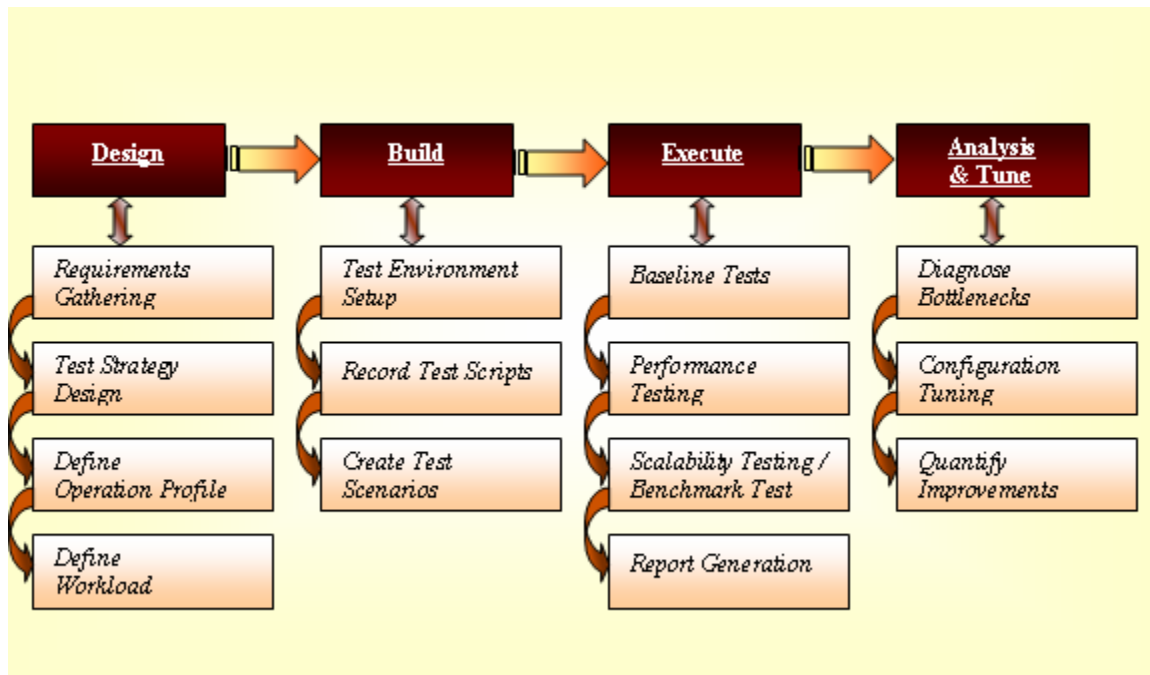
- **Test database creation/maintenance**
- **Load generation tools**
- **Resource monitoring**
- **Results analysis and reporting.**

### (f) Understanding of Architecture

To measure web application performance effectively and define what improvement it needs, it's quite important to take into account its architecture. This includes:

- **Web Browser:** the software client on which web applications run. It's independent of the application.
- **Internet Service Providers (ISP):** provide different speed and type of Internet access.
- The "**Big Internet**" is the world wide communication infrastructure between browser and server web.
- A **Firewall** is the interface between company intranet and Internet. They can filter the ingoing and outgoing traffic according to some rules defined by the web administrator.
- Other network elements can be found on the boundary between intranet and Internet: **proxies and reverse proxies**. Proxies help an intranet client finding an external site and besides they have the important function of caching, keeping in memory the most requested pages from the users of LAN. **Reverse-proxies** mask intranet internal servers providing to Internet some sort of virtual address that is forwarded to the right server.
- **Web Servers** is the applications able to meet requests from clients (browsers). It forwards the requests to the web-application that can be on the same machine or on another server devoted for that purpose (application server).
- **Application server** is the machine where the code of the applications runs. This machine can coincide with the web server, but if a company has many applications is better to allocate them on different servers, letting the web server the function of interface.
- **Database** holds the data of the applications. Access data can heavy. When they manage many data, access time could be too high. For that reason it's better to allocate a machine just for this function (DB server).

## PHASES OF PERFORMANCE TESTING



### (a) DESIGN PHASE:

It involves defining the business processes to be tested, the business process mix of an average or peak production hour, and the overall user and response time goals for the system

#### (i) Requirements Gathering

This activity is carried out during the business and technical requirements identification phase. The objective is to understand the performance test requirements, Hardware & Software components and Usage Model. It is important to understand as accurately and as objectively as possible the nature of load that must be generated.

Two types of Requirements needs to be gathered:

1. **Business requirements** are generally gathered by meeting with subject matter experts (SMEs). These may be business analysts and end users. A comprehensive set of business requirements exists when the following are in place:
  - **An Application Overview:** A demo of system usage to allow the performance team to understand, at a high level, how the application is used.
  - **A Business Process List:** A list of the key business processes that end users perform on the system.
  - **Business Process Flows:** Word documents that detail the exact steps/screens of each business process.
  - **A Transaction List:** A list of the key activities within the business process – such as Login or Transfer Funds – that need to be measured under load.
  - **Business Process Diagrams:** Business process flowcharts that illustrate branching conditions in the business process flows.

2. **Technical requirements** can be gathered by meeting with the system administrators and database administrators (DBAs). These folks may be part of Development, Operations, or both. A comprehensive set of technical requirements exist when the following are completed:
- **An Environment Walkthrough:** A walkthrough from the systems or infrastructure team on the testing architecture.
  - **Systems Scope Meeting:** A meeting to discuss and agree what pieces of the system will be stubbed out or excluded from the test process.
  - **Production Diagram:** A diagram of the production infrastructure to flag deltas that may impact performance during the migration from QA to production.

Following are the important performance test requirement that needs to be captured during this phase.

- Response Time
- Transactions per Second
- Hits per Second
- Workload
- No of concurrent users
- Volume of data
- Data growth rate
- Resource usage
- Hardware and Software configurations

### **(ii) Test Strategy Design**

Test strategy helps to decide what must get done and when. The strategy addresses the following concerns:

- What will and will not be tested and how?
- What priorities will be applied to the Scenarios that are identified for testing?
- What all Performance attributes would be captured during testing?
- What would be the time frame for testing?
- What would be the methodology used for Diagnosis of Problem & Fixing of the same?
- What would be the Performance Attributes that would be captured?
- What would be the Configuration for Test Environment (Hardware, Software, Database etc)
- Who all would be the resources allocated to the Project?

### (iii) Define Operational Profile

Operation profile is nothing but the business processes that would be carried out through the application. Operational Profile can be decided upon from the following information received during the Requirements Gathering Phase:

- List of Business Process that end-users perform
- Sequence of the Business Process
- List of Transactions that would be performed in each of the Business Process
- Sequence of the Transactions in the Business Process.

Based on this data, the Operational profile can be created which would give the following information:

- Sequence of the transaction
- Transaction Name
- % usage of the transaction

Sample Operation Profile:

Sequence	Transaction Name	Operation Profile%
1	Register User	10%
2	Online Shopping by Registered User	20%
3	Online Shopping by Visitors	30%
4	Search Products	40%

### (iv) Define Workload

Workload would be defined based on the business process, transactions and the number of users information got during the requirements gathering. Detail level workload would be defined for each of the business process/ operation profile that would be executed during the performance testing. Moreover, it would also include information as to what all performance attributes would be tracked during the execution.

Sample Workload:

Transaction Name	Number of Users	Measurements
Register User	1, 5 & 50 users	Web Server (CPU utilization) Response time, throughput, Latency
Online Shopping by Registered User	5, 20 & 60 users	Response time, throughput, Latency
Online Shopping by Visitors	10, 50 & 100 users	Response time, throughput, Latency
Search Products	20, 60 & 200 users	Response time, throughput, Latency

**(b) Build PHASE:**

It involves setting up and configuring the test system and infrastructure, and using the automated performance testing solution to build both test scripts and load scenarios.

**(i) Test Environment Setup**

Environment setup consists of assembling the hardware, software, and data required to execute a successful, realistic load test. This may involve working with the Systems, DBA, Operations and Business teams. It also includes deciding which servers or machines to monitor under load.

**(ii) Record Test Scripts**

It includes recording the documented business process into automated scripts. Following are the elements that need to be incorporated into the scripts:

- Transactions: Insert timers to produce the logical timings desired by the business
- Parameterization: Replace all input data such as LoginIds and passwords with a pool so each virtual user accesses the application using unique data.
- Delays
- Checkpoints
- Synchronization Points

**(iii) Create Test Scenarios**

Create operation profile and workloads which have been defined during the Design Phase. This activity is carried out by assigning groups of users different scripts, connectivity and user behavior.

**(c) Execute PHASE:**

It consists of running the load scenarios and measuring system performance.

**(i) Baseline Tests:**

This test verifies that the system and its surrounding environments function within reasonable technical parameters. Performance tests are run with only five to 10 users to baseline end-user transaction performance. These tests should be executed at the start and end of the performance testing process to measure absolute response-time improvement.

**(ii) Performance Tests**

**(iii) Scalability Tests**

**(iv) Benchmark Tests**

**(v) Report Generation**

**(d) Analysis & Tune:**

These iterative phases go beyond measuring system performance and take load testing to another level. The focus is on pinpointing problems to help resolve them rapidly and tuning system parameters to maximize performance.

- (i) **Diagnose Bottlenecks**
- (ii) **Configuration Tuning**
- (iii) **Quantify Improvements**

**References:**

1. Whitepaper on “*Automated Performance Testing*” by Mercury Interactive.
2. Whitepaper on “*Key Considerations in Performance Testing*” by Leslie Segal  
President Testware Associates, Inc.
3. Whitepaper on “*Performance Testing Challenges and Approach*” by Mahesh S
4. Whitepaper on “*Web Performance Testing and Measurement: a complete approach*” by G. Cassone, G. Elia, D. Gotta, F. Mola, A. Pinnola