



# Performance Test Workload Modeling

---

## Abstract

A few years back when there wasn't much use of web applications and the competition was very low, web applications' stakeholders and users were not greatly concerned about their non-functional features. But today the scenario has changed as we are living in a World Wide Web era where almost everything is over the web. While browsing over the web in order to find out any service information, you end up with countless providers who are just few clicks away from you. This growing competition also demands better customer services to get the competitive edge. Main purpose of web applications is to facilitate the users in getting their desired information in a quick and efficient manner. This makes web applications' performance the most important parameter to attract and retain maximum users. All critical applications' performance testing is conducted to validate whether they are meeting users' expectations or not and also to mitigate the identified performance bottlenecks. Simulating the application under test (AUT) real users' behavior is the fundamental point in performance testing. A workload model is designed to identify key test scenarios and load distribution across these scenarios. Performance test results can never be reliable unless its workload model is designed accurately. Identification of key test scenarios and load distribution across these scenarios is one of the major parts of a performance test plan which lies under the workload modeling. In performance workload modeling phase, a complete understanding of AUT is developed for multiple purposes such as to identify its test objectives, to figure out all its users' types and their activities, to identify application key business scenarios along with their navigation paths and eventually to analyze and evaluate the load distribution across all identified scenarios and their navigation paths.

In this white paper, we will learn about workload modeling, its importance, the detail of activities involved in designing the performance test workload model and the various challenges which the performance testing teams face while designing the workload model.



## What is Workload Modeling?

Load distribution across all the identified scenarios of AUT is called the Workload. Workload model of an application depicts how this application will be used in the production environment. In order to get such performance test results which portray the true picture of AUT, it should be tested on a workload model which is very similar to the production environment.

The complete understanding of AUT and its usage patterns is mandatory before modeling its workload. For a better understanding of the concept, workload modeling can be divided into business workload and infrastructure workload.

### Business Workload

Set of user actions or business scenarios performed within the test to achieve the business objectives and goals is called the business workload. A good performance analyst must have a complete understanding of all the application major business transactions (which will be included in the performance test) and their impact on the AUT overall performance. In one of our previous white papers ([Load Testing Scenarios Selection](#)) we have discussed in detail the practices that could be followed to identify and pick the right set of AUT performance scenarios.

### Infrastructure Workload

The amount of resources/infrastructure used by the business scenarios to achieve their desired goals is called the Infrastructure workload. CPU, memory and IO operations utilization during the test is an example of infrastructure workload model which you set while defining your test goals.

## Importance of Workload Modeling

Your performance test results will be more accurate if you manage to properly simulate your production system parameters in your test. It's the planning phase where the performance analyst makes sure that the information of all the parameters of AUT has been acquired in order to accurately simulate them in the performance test. Identifying AUT workload model is one of the most important parts of the planning activity. Workload model provides the information of what type of user actions will be tested under load, what will be the business scenarios for all the users and what will be users' distribution on every scenario. This information helps the performance testing teams in many ways such as:

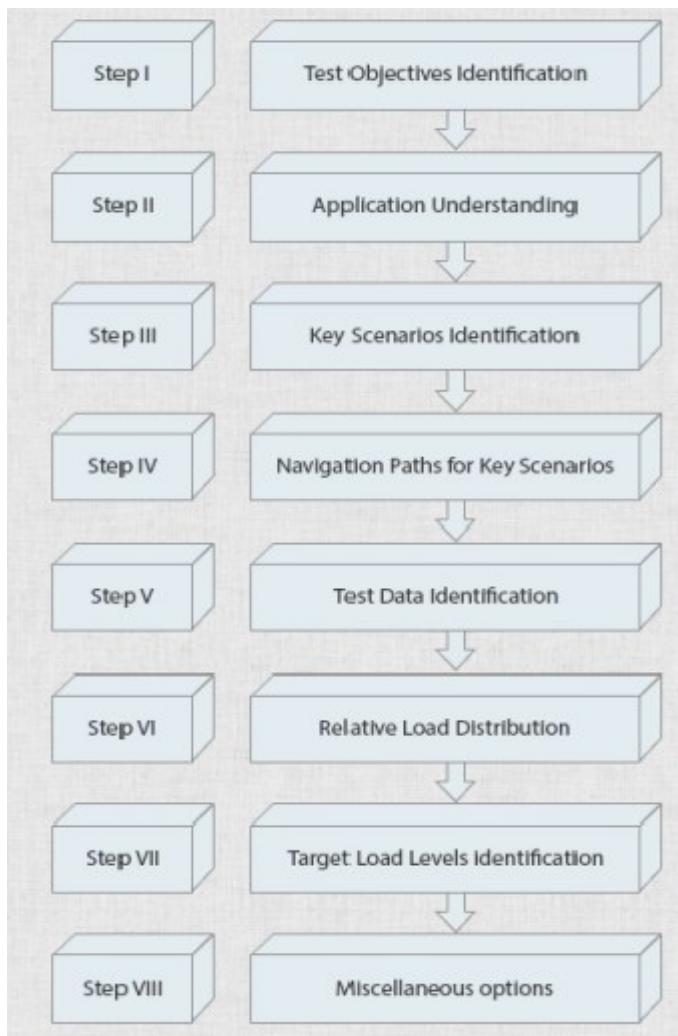
- **Performance Scenarios Identification:** The fundamental activity of the Workload model is to understand the application and identify its performance scenarios.
- **Performance Test SLAs:** Performance testing teams translate AUT non-functional requirements into performance test SLAs through workload 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.

### Activities involved in Workload Modeling

Performance testing is a complex activity as it consists of various phases and each phase has several activities in it. Workload modeling is one of the most important parts of the performance testing activity and it's not simple by any means. Some of the activities necessary for identifying the performance test workload model are listed below:





## 1. Test Objectives Identification

Not just the performance testing but in any activity you put efforts which are aligned to your objectives. Identifying your test objectives means to examine what actions you need to take in order to successfully achieve those test objectives. So before we formally start working on any application's performance testing, first step is to identify its test objectives in detail. For an E-commerce web application, following are some of the examples of performance test objectives:

- **Response Time:** Product search should not take more than 3 seconds.
- **Throughput:** Application server should have the capacity of entertaining 500 transactions per second.
- **Resource Utilization:** All the resources like processor and memory utilization, network Input output and disk input output etc should be at less than 70% of their maximum capacity.
- **Maximum User Load:** System should be able to entertain 1000 concurrent users by fulfilling all of the above defined objectives.

## 2. Application Understanding

Complete understanding of the AUT with all its features is the basic step in any testing activity. You can't thoroughly test an application unless you have its complete understanding. Same is the case with the performance testing. Performance testing starts with planning and planning starts with application understanding. You explore the application from performance perspectives and try to get the answers of the following questions:

- How many types of users are using this application?
- What are the business scenarios of every user?
- What is the AUT current and predicted peak user load for all its users' actions over time?
- How the user load is expected to grow with time?
- In how much time a specific user action will achieve its peak load?
- For how long the peak load will continue?

Performance testing teams can also collaborate with network team to check the server logs to get the answers of some of the above mentioned questions. They can also interview the marketing team and application stakeholders for some of the answers.

## 3. Key Scenarios Identification

It's neither practiced nor required to simulate all user actions in performance tests due to the budget and time constraints. Performance testing teams always select limited number of user actions which have greater performance impact on the application. In one of our previous white papers ([Load Testing Scenarios Identification](#)) we have discussed this topic in detail. Following are the examples of few scenarios which should be selected while conducting the performance tests:



- **Measureable Scenarios:** A basic criterion for selecting any user scenario for performance testing is it should be fully measureable.
- **Most Frequently Accessed Scenarios:** Application scenarios which are mostly accessed by the users when they browse through the application.
- **Business Critical Scenarios:** Application core scenarios which contain its business transactions.
- **Resource Intensive Scenarios:** User scenarios which consume more resources as compared to typical scenarios.
- **Time Dependent Frequently Accessed Scenarios:** Such application scenarios which are accessed on specific occasions only but are accessed very frequently.
- **Stakeholders Concerning Scenarios:** Application features about which the stakeholders are more concerned such as AUT newly integrated modules.

Some of the most desired performance testing scenarios of an E-commerce application could be,

- Browsing product catalog
- Creating a user account
- Searching for a product
- Login to application
- Order Placement

#### 4. Determining Navigation Paths of Key Scenarios

Once you have identified all the AUT scenarios which should be included in a performance test, next step is to figure out each scenario's all possible paths which a user can opt to successfully complete it. Any application users most probably have different level of domain and technical expertise and it's quite obvious that they will follow different steps to complete a specific scenario(s). Performance testing teams identify all possible paths which users could follow to successfully complete the identified scenario and also figure out the frequency of each path to decide whether it should be included in performance test or not? Application response for the same scenario can greatly vary depending upon user navigation path and it's strongly advised to test the selected scenario's all major paths under load. Following are the few guidelines which could be followed to identify any scenario's navigation paths:

- Figure out AUT paths which can be used to successfully complete more than one identified scenarios and have major performance impact.
- Read manuals (design and user) to find out identified scenario's all possible paths.
- In case of production application, check log files to find out the users' navigation patterns to complete an identified scenario.
- Explore the application and try to find out all possible paths of a scenario by yourself.
- Another approach could be to provide application access to new and experienced users and ask them to complete certain scenarios and observe their actions.



## 5. Identify Unique Test Data

Unfortunately, identifying selected scenario's all possible navigation paths doesn't provide all the information required to successfully simulate them in a performance test. Several bits of information is still required to accurately simulate the workload model and preparing the required test data is one of them. You can never achieve accurate test results with improper or inefficient test data. One can develop an initial idea about the required test data by getting answers of the following queries:

- While navigating through a specific path, how much time a user spend on a page?
- Which conditions force the user to alter the navigation path?
- What input data should be required for a specific path?

Following table will show you an example of unique input data for browsing product catalog scenarios of an E-commerce application

Scenario	Action	Input Data	Output Data	Think Time
Browsing Product Catalog by an existing user	Login	<ul style="list-style-type: none"><li>• Unique username</li><li>• Password of the username</li></ul>		5-8 seconds
	Browse	<ul style="list-style-type: none"><li>• Catalog Tree</li><li>• User Type</li></ul>	<ul style="list-style-type: none"><li>• Product description</li><li>• Title</li><li>• Category</li></ul>	4-30 seconds
Browsing Product Catalog by a new user	Login	<ul style="list-style-type: none"><li>• Unique username</li><li>• Password of the username</li></ul>		5-15 seconds
	Browse	<ul style="list-style-type: none"><li>• Catalog Tree</li><li>• User Type</li></ul>	<ul style="list-style-type: none"><li>• Product description</li><li>• Title</li><li>• Category</li></ul>	10-60 seconds

You will also require a greater amount of test data and you need to keep an eye on the data base state if you wish to effectively run the test. Following are the few considerations which should be followed while executing a performance test where multiple navigation paths are tested for identified scenarios:

- Make sure that you have all the required test data as you will need more test data when you are testing scenarios with all navigation paths.
- Avoid using same test data for multiple users as it will produce invalid results.
- Periodically test the database status during the test execution and make sure it's not overloaded.
- Include some invalid test data as well to simulate the real users' behavior because users sometimes provide invalid values as well.

## 6. Relative Load Distribution Across Identified Scenarios



Now that you have understood the application, identified its key test scenarios, their navigation paths and test data required for each identified scenario, next step is to figure out the distribution of all the identified scenarios in your workload model. It's quite obvious that in any application few scenarios are executed more frequently as compared to others. Moreover, for some applications' scenarios the execution also depends on relative time. For example, for an online billing application where bill payments are made only during first week of the month, the administrator will be using the application mainly for accounts updating and importing billing information etc. in last 3 weeks of the month but as the first week starts, the major chunk of site users will be using it for bill payments. So you need to figure out which AUT scenarios will be executed at what time and what will be their execution percentage in the overall workload. Following techniques could be helpful in identifying relative distribution of your identified scenarios:

- Check out the server log files to identify users' trends on application if it is in the production environment.
- Consult with the sales and marketing teams to figure out which features they believe will be mostly used.
- You can also interview the existing and potential clients to find out in which features they are most interested.
- Another approach could be to share a beta release among a smaller segment of users and check out their trends and behaviors on application from server log files.
- In case none of the above mentioned approaches work then use your experience and intuitions to get the answers of your questions.

For example for an e-commerce application, the identified scenarios distribution could be as follows:

User Scenario	Percent Load Distribution
Browsing product catalog	40
Creating a user account	05
Searching for a product	30
Login to application	15
Order Placement	10
<b>Total</b>	<b>100</b>

## 7. Identify Target Load Levels

One last step involved in workload model completion after performing all the above mentioned activities is to identify the normal and peak load levels of every selected scenario to test the application for expected and peak load conditions. Mainly depending upon the application you need to identify the monthly, weekly, daily and hourly average load targets for every selected scenario. You need to know the following information in order to identify the target load levels for an application:

- What are the current and expected normal and peak user requests level?



- What are the application key test scenarios?
- What is the distribution of user requests on the identified scenarios?
- What are the navigation paths for all the scenarios along with relative utilization of every scenario?

Following table will summarize what kind of information you need while identifying the target load levels for any scenarios

Time Period	Normal Load Requests (Avg.)	Peak Load Requests (Avg.)	Peak Load Build up Time	Peak Load Duration
One Month	72,000	2,16,000	2 Hours	1 Hour
One Week	16,800	50,400	2 Hours	1 Hour
One day	2,400	7,200	2 Hours	1 Hour
One Hour	100	300	2 Hours	1 Hour

## 8. Workload Design Miscellaneous Options

Once you have identified the key test scenarios, their relative distribution and the target load levels the last thing is to figure out different options for your workload like browser mix, network mix, think time and pausing between the iterations to simulate the workload as per the real environment. All performance testing tools are equipped with following options:

- **Browser Mix:** List down all the browsers which you want to include in your test and specify the load distribution across all browsers to verify what will be the system response in all these browsers.
- **Network Mix:** There are various internet connections available these days and people use almost all of them based on their availability and constraints. So it's better to include major list of network connections in your test and appropriately distribute the load on them.
- **Think Time:** As real users always take some time while taking different actions, it's very important to include the think time in your test based on the application users' comfort level in using the application identified scenarios. Ignoring the think time can invalidate the test results.
- **Pause Time:** There is always a certain pause before the user receives a response from server and send a new request which can be cater to with pause time between the iterations.

## Challenges Involved in Workload Modeling

We have discussed all the activities involved in workload modeling and its quite obvious that the workload modeling for performance test is not a piece of cake. Performance testing teams face various challenges while performing all the above mentioned activities. Following is a list of some of these challenges:

- Complete application access in planning phase before executing the performance test





- Getting help from the marketing and network teams for their input on application performance critical scenarios and their workload distribution
- Accessibility to relevant server logs
- Applying the data mining techniques on server logs to extract the relevant information
- Presenting the workload models to the stakeholders in an effective manner to get their approval on it.

## **Conclusion**

Web applications' performance is becoming extremely important due to the growing competition. Today, application stakeholders are more concerned about their applications performance and are not reluctant any more for conducting the performance testing activity on them. Simulating real users' behavior in your performance test is the key to achieve the desired performance test results. Designing a performance workload model very similar to the AUT production environment is one of the core activities in performance testing. A workload model will provide you with the list of parameters (types of application users, their key business scenarios, navigation paths of each scenario and load distribution across all scenarios etc.) which you need to accurately simulate in your performance test in order to achieve the desired results. Getting all the required information for developing an accurate workload model is a challenging task and performance test teams often collaborate with the application stakeholders, network and marketing teams for obtaining this information.