Abstract

Crystal clear requirements before starting an activity are always helpful in achieving the desired goals. Achieving desired results are quite difficult when there is vague or incomplete information and situation get even worse when source of information is not known. Same is the case with performance testing where complete test requirements prior to the activity are extremely important for its success. This paper is all about collecting the performance testing requirements before starting the test activity when minimal or no information is available. First we will understand what elementary information is required to initiate the performance test and then we will discuss different approaches and techniques to collect the relevant information from the concerned sources. We will review a structured approach that will guide reader to know the order in which he/she needs to contact the sources. Furthermore, we will share a list of questions to collect each segment of information as well as strategy of deciding about performance parameters in the absence of any help.

Introduction

The quality of service attributes like performance, availability and reliability are becoming more and more important in this fast competitive technological era. You can only survive in the market by providing best services at a rapid pace. Today, performance is one of the key reasons of software failures. Studies revealed that unavailability or ambiguity in performance requirements are the core reasons for more than 50 percent of poorly performing applications. Businesses have started realizing the importance of performance testing and they are investing enormous efforts in it. Regardless, performance testing process is not strictly followed as most of the times useful information is not provided which eventually leads to unproductive results.

"*Performance testing is only as accurate as the model you simulate – time invested in the requirements is time well spent*" – Jason

In a testing activity, ample time is spent on project execution while less attention is given to the performance requirements gathering. The activity is performed against specific needs which establish test foundation. Unlike functional testing, where you can achieve decent results even with ad-hoc testing however clear requirements are mandatory for successful performance testing.

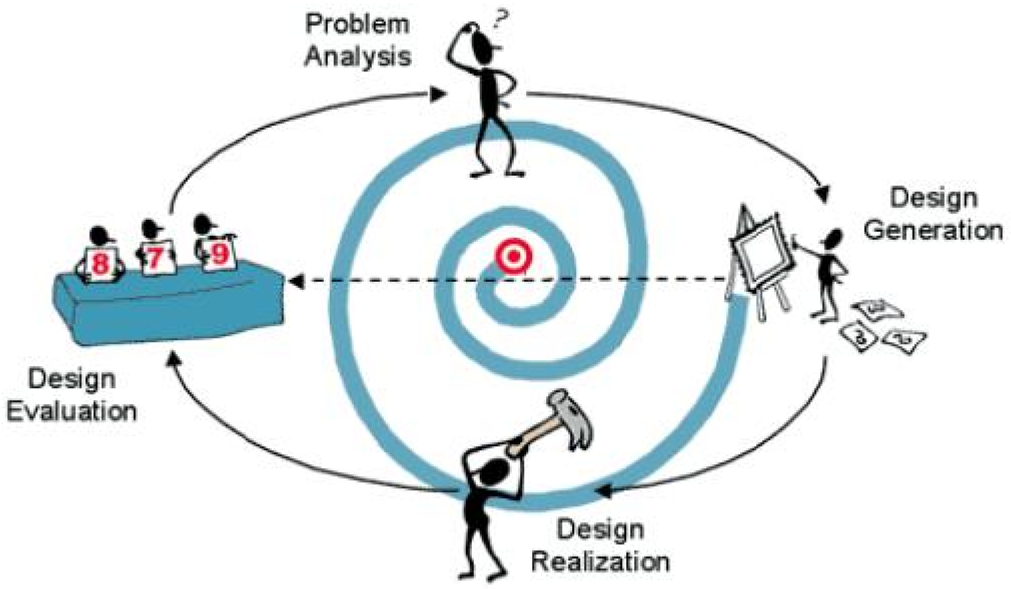
Now we will discuss various strategies to gather performance requirements before commencing the actual activity.

WHAT WE NEED TO START THE PERFORMANCE TESTING?

It is essential that one must know what he/she needs to start the testing activity. Here is the list of all the requirements pertaining to the activity however there is no rule of thumb.

Few questions flash in mind immediately whenever an application is given for performance testing. These questions are as following:

- What is the type of application and its architecture?
- What are the known current as well as previous performance bottlenecks?
- Which application scenarios to be tested?
- What will be the workload model?
- What are the performance goals?

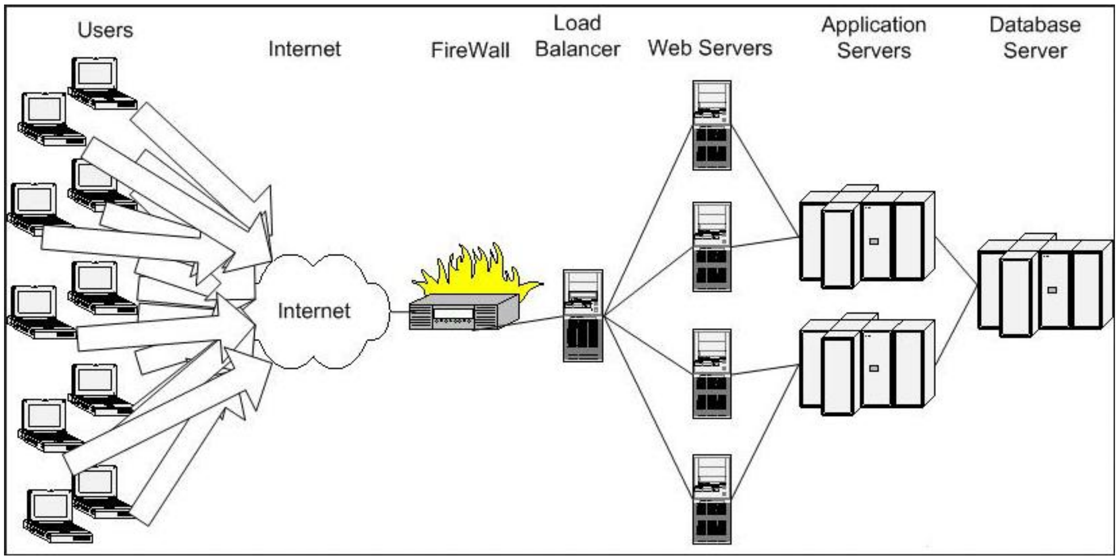


Complete and clear non-functional requirements in the shape of performance test plan are vital to answer the above questions. Without getting the answer of these questions, a successful testing activity is not guaranteed. All these points will be discussed here in details.

APPLICATION TECHNOLOGY AND ARCHITECTURE

Application type (web, desktop or mobile) and its development technology information are always crucial to make a decision on appropriate performance testing tool. Performance testing is now automated and there are number of tools used to for it. Different tools come with different features and one can't start testing until he/she has full access to that particular tool which completely supports application under test (AUT).

Further, complete AUT architecture information including all the associated components (like Web, Application and Databases servers, Network bandwidth etc.) must be known in details to setup right test environment.



Setting up a testing environment, which should be replica of the production environment is most important and fundamental aspect in performance testing and results will be based on that environment. Accurate performance results can never be achieved unless the production environment is not replicated.

SETTING UP THE TEST ENVIRONMENT

In many cases performance testing team is asked to test the application in production environment. It is the responsibility of the person who is performing test to guide all the stakeholders on risks associated with testing in production environment. For example, actual users working on application can impact the performance results. Simultaneously, tests can also affect the real time user's activities and application can be unresponsive.

So it is always recommended to test the application in a controlled environment separate from the production environment. In order to setup a test environment similar to production environment, complete information of live environment is essential.

Network team is the primary contact to get complete production environment information. They must have complete information of the application architecture and its components. They should be able to help in setting up test environment. In case it is not available from network team, secondary contact will be development team.

Following question can asked to collect desired information:

- What is the application type?
  E.g. Desktop, Web or Mobile App or any other
- In which technology/platform the application is developed?
  E.g. J2EE, .Net, PHP, Silverlight, Ruby, SAP, Any other
- Which data base is used?
  E.g. Oracle, MySQL, SQL Server
- Which Application server is running with the system?
  E.g. Tomcat, IIS, WebSphere
- How is the targeted application look like? (Please specify all servers and network appliances configurations and their interaction mechanism)
  - LAN/WAN details
  - Terminal servers
  - Bandwidth link
  - Load Balancing techniques
  - Batch Transactions
  - Disaster recovery
- What is the protocol between the client and server?
  E.g. HTTP, HTTPS, FTP, TCP/IP, Telnet etc.
- Is the client browser version dependent in case of web application?
  E.g. Application runs only on IE-8
- Will separate test environment be provided to do a performance test run?
- Any preference on performance tools?
  E.g. LoadRunner, JMeter
- What is the current project lead time for testing activities?
  I.e. starting and completion date

CURRENT AND PAST ISSUES

Once the application information is collected as discussed in previous section, next step is to decide which test needs to be applied to catch the performance bottlenecks. The required performance test(s) can be identified with the help of known current and previous performance issues of the application. Moreover, this information will establish the goal of the performance activity. There can be various performance issues likes more response time, memory leaks, lower throughput, unexpected growth in daily visitors, high CPU utilization etc.

There are various types of performance tests like load test, stress test, soak test, volume test, scalability test etc. and each test is used to find out specific type of performance issues. Therefore, detailed information of all the performance issues will help the test team to select the right type of performance test.

Production Environment

Let’s review how to get information of current and previous known issues when the application is in production environment. As discussed in previous section, development and network teams will be in a better position to provide the required information. Though, test team can also be consulted for their input. Also, issues status must be known if they are resolved or not.

Pre-Production Environment

On the other hand, if application is not yet in production environment, functional team can be asked about various bugs they would have encountered during testing. Their input would also help in deciding the type of performance test.

Following question can asked to collect desired information:

- Is the functional testing completed?
- Are all functional testing defects fixed?
- Is there any incomplete development of application feature(s)? Those must be identified
    - Specify the functionality
    - Specify current behavior
    - Specify desired behavior
    - When will this be fixed?
- Is there any known issue(s) in this application? E.g.
    - Memory lock
    - Unexpected growth in daily visitors
    - More response time which leads to time out error
- What type of performance tests need to be performed? E.g.
    - Load Test, Stress Test, Soak Test, Spike Test, Scalability Test
- What are the goals of the performance testing activity? E.g.
    - Evaluate System against performance criteria
    - Discover what parts of the system perform poorly and under what conditions
    - Compare two platforms with the same software to see which performs better

Performance Scenarios

As discussed in previous sections, since appropriate test environment is prepared similar to production environment and required performance test type is identified, now what is the next step? It is time to identify the user transactions that need to be simulated in performance test. Identification of right performance scenarios is extremely important for successful test results. The selection of wrong user transactions can cause skips in many of performance bottlenecks which will reveal in production environment and complete performance testing activity can be wasteful.

The performance scenarios are different from functional testing scenarios where 100% application coverage is not required. It will be an infinite loop if all application functional scenarios are selected for the performance test. Similarly you may miss many critical bottlenecks by considering only few of them.

80-20 rule (20% of the transactions cover 80% of application's core functionality) is normally applied while selecting the performance scenario. Those transactions are selected which will be mostly accessed by the users. But based on the importance of certain transactions you also select them even when they don't follow the famous 80-20 rule.

Now we have got the understanding of performance scenarios and their importance as well, let's start hunting them. Different methods can be applied to identify performance scenarios for the applications in production and pre-production environments. Let's discuss them one by one.

Pre-production Environment

It's bit challenging to identify required user transactions if the application is not in production. In this case, the first point of contact should be the person involved in application planning. Normally, business analyst and product marketing teams have this information as they document which application areas are most important from end user's perspective and will be accessed extensively. If you can't find required information from these sources then product sales representatives will be your next point of contact. These people directly interact with potential users and always have some idea of product features that will be of great user's interest and those will be your performance scenarios.

Production Environment

If the application is live and being used by end users, its performance scenarios identification will be an easier task compared to pre-production. You will start exploring by contacting same sources so ask business analyst, marketing and network teams for the required scenarios. In production environment usually different web analytic tools are deployed on servers to monitor end users activities. These tools provide you the required information like which application components are mostly being accessed by the end users and what is the frequency. Network teams have full access to the monitoring tools and these statistics are easily available to help you to select the appropriate user test transactions. Similarly you can also reach out to product sales team if you don't find any fruitful information from above mentioned sources.

If you are not lucky enough to find required information from any of the above sources, the only option is to utilize your own experience to figure out performance scenarios. Analyze the application from end user perspective to identify the transactions user will be most interested in. Being a good performance tester one must have good analytic skills and the ability to analyze the application from end user perspective. Following type of application transactions can be selected for the performance test,

- Business critical transactions
- Most frequently accessed transactions
- Performance intensive transactions

For example if you are going to test an e-commerce application, order placement will be the business critical transaction, browsing catalog will be the most frequently accessed transaction and searching a product will be the most performance intensive transaction and all of them should be selected for the performance test.

Following question can asked to collect desired information:

- Have you identified the performance scenarios of the application?
  E.g. e-commerce application performance scenarios can be order placement, browsing catalog, product searching etc.
- Have you deployed any traffic monitoring tool on server?
  E.g. Google Analytics, New Relic
- Are there any inter process dependencies that affect the user activities?

WORKLOAD MODEL

As we have discussed various methods to identify the AUT information, its current and previous issues and performance scenarios, next step is to find out the number of users that will be simulated and their right mix for each of the selected transaction. User distribution on selected scenarios according to the production environment is extremely important to completely test selected application. A performance test executed without selecting the right workload model is never guaranteed to depict the true performance results.

This workload and its distribution will also help in knowing type and amount of test data needs to be prepared. Preparing the test data is a pre-requisite of performance testing and every transaction should be tested with similar data values used by actual users. Moreover, system should have same amount of data in the database prior to the testing to properly find out memory or database bottlenecks.

Another important aspect of identifying maximum user count is to identify number of load generators you will be needed to test the application. Load generator is normally a separate component associated with almost every load testing tool. These load generators are normally installed on separate machines and are used to simulate users in the test. There is always limited number of users that can be simulated from a load generator depending upon the application technology and system specification on which the load generator is installed.

Now how to gather this workload information? The techniques we discussed above for selection of user scenarios will be used for finding the workload as well. Again you can ask a set of questions from business analyst and marketing team to find out the system workload model. If you don't get your answers, you can ask the network team to check the web server logs to find out the users distribution across different application areas. Network team might be able to provide the user distribution but they will not be able to provide the maximum users or TPS against which the application should be tested. In this situation, you will use your experience to start application testing in incremental manner to reach a point where bottlenecks will start producing or application crashes.

Another way to find out workload is to use a formula known as "Little Law". This formula is widely trusted to find out maximum number of users in a performance test. One can use this formula if he/she has baseline throughput (i.e. TPS) and response time associated with a user. But the real challenge is to get TPS and response time information to use this formula.

Following question can asked to collect desired information:

- If selected scenarios require some unique inputs then these should be specified.
  E.g. Credit Card, SSN etc.
- Do you have statistics, how many users visit the application in 24 hours?
  E.g. Facebook is access by more than 175 million users daily
- What is the peak load time on production server?
  E.g. Maximum number of US based users log on to facebook.com at 8pm EST
- How many users access the application in peak load time?
  E.g. Facebook is accessed by up to 10 million users during peak hours
- What is the average user session time on application?
  E.g. Facebook user average session time is 23 minutes
- What will be user distribution on test scenarios?
  E.g. on Facebook, 1 million users will concurrently login, 4 million will view posts and 1 million will add posts etc.
- How many users are intended to access application simultaneously?
  E.g. Currently Facebook is supporting 10 million users simultaneously but in future it would support 20 million users
- Are there any time constraints for running the test?
  E.g. the server can only be accessed outside business hours; server can only be accesses from 7 pm – 8 am
- Is it required to generate load from multiple geographical regions? If yes, which?

PERFORMANCE GOALS

Since all methods to identify application architecture, test type(s), performance scenarios and workload model have been discussed; last step is to figure out the performance goals against which each test result will be evaluated. These performance goals are used to define the test pass/fail criteria.

These performance goals will help in decision making whether the application is ready for production or not. There are various parameters which can be selected for the pass/fail criteria of the performance test. Some of them are as following,

- Response Time (E.g. search should not take more than 3 seconds)
- User load (E.g. application should be able to handle 500 concurrent users)
- Transaction Rate (E.g. application must be able to handle 50 transactions per second)
- Hardware Resource Utilization (E.g. CPU utilization on application server should not exceed 70%)

Any of these factors or their combinations and various other factors can be used as performance goal. More detailed performance goals more realistic results will be.

Now the primary question that comes to mind is how to identify these performance goals. The procedure will be the same applied in above requirements. Various stakeholders like business analysts, marketing team, network team etc. should be contacted and they should be able to provide the required information to great extent.

If these sources are note able to assist in any particular aspect, then use your expertise and do some brainstorming to identify the right set of performance goals and their acceptance criteria. There is no specific standard of each performance goal and they change from time to time, situation to situation. For example, a few years back, 8 seconds was assumed as the good response time of an application but today if an application not responding within 3 seconds is normally considered below performance. Similarly, critical application response time will be far less than the other application. So it’s important to thoroughly analyze the application to identify associated performance goals and their values.

Following question can asked to collect desired information:

- Have you set any acceptable maximum transaction completion time?
  E.g. System response time should not exceed 3 seconds while retrieving user's order history
- Have you set the expected throughput of the application?
  E.g. 1000 transaction per minute
- Which is the most important performance goal of the application?
- What will the acceptance criteria for each performance test?
  E.g. all user transaction should pass with response time less than 3 seconds and CPU utilization should be less than 70%

CONCLUSION

By concluding this discussion, we can say collecting clear performance testing requirements prior to the activity are extremely important for successful outcomes. Unfortunately, most often limited information is provided and finding the required information is an uphill task unless one exactly knows what is needed and how to gather it. However, by applying the methods and approaches discussed above, one can better hunt for performance test requirements by asking right questions from concerned personnel. Even if no response is received from these sources, best approach is to use your own experience and expertise.

Sajid Manzoor