**How to run performance test on mobile apps**

Performance testing should be conducted on mobile applications just like the way it is conducted on any conventional web application or any other enterprise application. However the type of testing is same the perception that a software tester possess must be different, given the way the application is designed and is used in real world.

**Performance testing**

In a nutshell performance testing can be explained as, a category of non-functional testing which validates the behavior an application under a desired load. Mobile application shouldn’t be exempted from the this test to prove the below requirements

* To ensure it responds to the user without any delay during peak traffic
* The application is scalable as per the load
* Application able to handle the concurrent requests from different users
* More importantly the application doesn’t get crashed under bottleneck conditions

**Things that make testing of a mobile app is different than normal application**

In general if we take any conventional web application performance test strategy/ plan, performance tester would be given some nonfunctional requirements (NFR) and below are the some of the common objectives

* Subjecting the application to a specific load
* Hitting the application with a specific number of users
* Injecting the lot of volume of data

The above discussed points are pretty muc business requirements, when analyzed how technically those above tasks are achieved, how those are actually performed on the application, we understand that they are as below

* Hitting the api continually for a specific time to perform certain business logic and measure throughput of the server
* Traversing through the UI screens and measure the response time
* Placing and retrieving some records from the message queue

Now, when it comes to the mobile testing we have lot of factors that come into picture, below are some factors that proves mobile performance testing is little different.

**Mobile application types**

Mobile can typically run 3 different kinds of applications, so there should be difference in way we test them, the types are as below

1. Web applications
2. Native applications
3. Hybrid applications

Each application has to be tested in its own strategy, above that we have quite a few o mobile types on which the testing needs to be performed, let’s look at each one of them

**Web application**

These kinds of applications are only accessed from the mobile browser window. Hence the possible scenarios that come under performance testing are,

* Accessing the application page concurrently with many users and see if the performance is acceptable.
* Traversing through the different tabs and screens present in the application and measure its response time.

We can make use of simulators/ emulators to test this, as we are only concerned about application performance on the browser despite the mobile.

**Native application**

An application is said to be native application if and only if, it operates on the mobile operating systems. The possible non-functional requirements would be

* Measure the CPU utilization
* Memory utilization when in run
* Consumption of the battery
* Response time of the application to the inputs

In order to test these requirements we must procure hardware devices because these are very specific to hardware.

**Hybrid applications**

These applications can be accessed via browser as well as can be installed on a device. The performance strategy would be a mix of both the above requirements we discussed.

**Various mobile device and their hardware standards**

As we just discussed, in order to test the native mobile applications, we must procure different mobile devices that are available. At least the devices whose usage is more in the market should be used for these testing.

**Network capabilities**

Since the mobile devices use the internet through the various network service providers, we tend to see a significant variance from network to network. Hence testing in this perspective to enhance the user experience makes much more sense

**Mobile performance testing strategy**

Any application that runs on production instance will look up a server to fetch/store the information in case of web, hybrid apps. On the other hand if the application is a native one, it might look up an in memory server and database to display and store the information

From the above context it is evident that there are two things involved here, one is client and other one is server, let’s understand them in laymen terms

**Client**

System that sends a requests to the server to receive the data and send data to store in the server

**Server**

System that actually responds to the clients request and provide data.

So, the mobile application performance testing is all about testing the client side system i.e. the application that runs on the mobile device and the server who provides data

**Application side parameters (client)**

**Application launch time**

How much time the application actually takes to get the application home screen displayed for the user is known as application start/launch time. It is also called as the response time. A good iteration of a this test should be conducted to ensure the application comes up quickly and there is no delay

We can actually replicate this case in a simulator and with different mobile types

**Application performance while multitasking**

This testing is conducted to measure the response time/ performance of an application when the user switches from this application to another one and comes back. This comes under an exploratory test case even though it is a good business use case. During this the application should not hang also it should be able to retain its previous state

**Memory consumption**

The application should be light weighted as possible and shouldn’t consume much of memory that it puts the whole operating systems in jeopardy. The memory consumption of the application should be very consistent.

**Server side parameters**

**Server throughput**

In a client-web server architecture the client will hit certain endpoints through HTTP/HTTPS protocol to get/save or delete information. It is equally important to test the server response time as well in sending the information. It is quite possible that many concurrent applications would be requesting the data from the server and sometimes for the same service, hence a performance test for a specific number of hours and with specific load should be conducted to validate the threshold performance

**Server scalability**

Scalability talks about how efficiently the server can scale up as per the increase of traffic. This can be achieved by increasing the number of threads that run on the server. This can be tested to understand the versatility of the server.

**Tools that are used for performance testing**

Below are the tools that we can use to performance test the mobile application, there are a few but listed the most popular once.

* Jmeter
* gatling
* Appium studio

**Performance testing best practices**

When we go through the testing process we experience lot of challenges and at the same we find equal amount of learning. It is always a good practice to capture the problem statement and the resolution for it, just to ensure we don’t spent the time over and over again on the same problem. In a similar way best practices are the lessons learnt from the past assignments .below are the some of the things which should be followed for better results

**Isolate the environments**

Unlike other levels and types of testing, performance testing must contain separate environment as we need to monitor the execution keenly. If at all we have shared environment between the different testing teams, we can’t deem the results very accurately. It is also very essential that database must be a separate one.

**Use simulators/ emulators**

The topic of using the real mobile devices or the emulators/simulators is legitimately a debate, but we should be mart enough in using the proper one as per need. Any characteristics of the web application such as network, page responsive time, load factors must be tested on a simulator/emulator to test them accurately and any native application performance characteristics, hardware dependencies can be tested using real device

**Use cloud solutions for more testing**

Cloud is one of the best solutions to use in testing. Because cloud saves lot of capital expenditure that we spend on on-premise infrastructure. We can use the shared resources i.e. servers, mobile virtual devices , databases, network setups..etc that are useful for testing. We can also use them as we want and pay as we use.

Thanks for the read, hopefully the blog conveyed some good knowledge about mobile application performance testing