

DIFFERENCE BETWEEN PERFORMANCE TESTING AND FUNCTIONAL AUTOMATED TESTING

For the last few months I have been working on performance testing quite a lot and when discussing it with colleagues I started to notice that it can be easily confused with test automation. Based on discussions I have had with customers and sales people I ran into the question of “what is the exact difference between the two? Both are a form of automated testing in the end”.

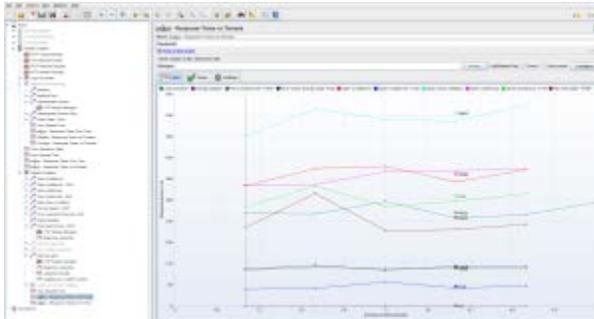
Performance testing == automated testing... ?

Both Performance testing and automated testing are indeed some form of executing simple checks with a tool. The most obvious difference being the objective of running the test and analysing the outcomes. If they are indeed so similar, does that mean you can use your automated tests to also run performance tests and vice versa?

What *is* the difference?

I believe the answer is both easy and challenging to explain. The main difference is in the verifications and assertions done in the two different test types. In functional test automation (let's at least call it that for now), the verifications and assertions done are all oriented to validating that the actual full functionality as described in the specification, was passed.

Whereas in performance testing these verifications and assertions are more or less focused on validating that all data and especially the expected data is loaded.



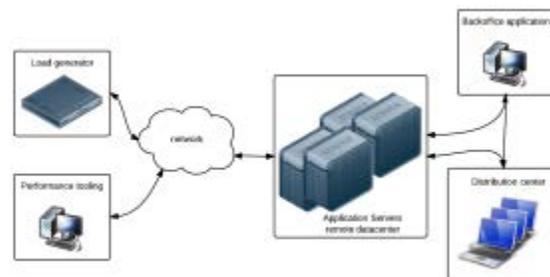
A lot of the performance tests I have executed over the past year or so, have not used the Graphical User Interface. Instead the tests use the communications underneath the GUI, such as XML, JSON or whatever else passes between server and client. In these performance tests the functionality of the application under test is still run through by the tests, so a functional walkthrough/test does get executed, my assertions however do not necessarily validate that and definitely not on a level that would be acceptable for normal functional test automation. In other words, most of the performance tests cannot (easily or blindly) be reused as functional test automation.

Now you might think: “So can we put functional test automation to work as a performance test, if the other way around cannot easily be done maybe it will work this way?”

In my experience the answer to this is similar as when trying to use performance tests as a functional test automation. It can be done, but will not really give you the leverage in performance testing you quite likely would like to have. Running functional test automation

generally requires the application to run. If the application is a webapplication you might get away with running the browser headless (e.g. just the rendering engine, not the full GUI version of the browser) in order to prevent the need for a load of virtual machines to generate even a little bit of load. When the SUT is a client/server application however the functional test automation generally requires the actual client to run, making any kind of load really expensive.

How can we utilize the functional test automation for performance testing?



performance and testautomation combined

One of the wonderful possibilities is combining functional testing, performance testing and load testing. By adjusting the functional test automation to not only record pass/fail but also render times of screens/objects, the functional test automation suite turns into a performance monitor. Now you start your load generator to test the server response times during load, once the target load is reached, you start the functional test automation suite to walk through a solid test set and measure the actual times it takes on a warm or hot system to run everything through a full rendered environment. This gives wonderful insights into what the end-users may experience during heavy loads on the server.

Source : <http://martijndevrieze.net/2013/08/28/difference-between-performance-testing-and-functional-automated-testing/>