Worqference 2023

Systematically start Web Load Testing using Apache JMeter

Instructor: Sandeep Garg

Frequently Asked Questions

1. What is Load Testing?

Amongst many other interesting and plausible definitions available on the Web, I prefer to say that, in Load Testing, ***Test scenarios*** are designed to ***simulate (mimic)*** the real-world ***concurrent usage*** of a software system using ***virtual users or threads*** and gather ***data*** w.r.t ***overall system health and user experience.***

The simulation is often done using a specialized tool like JMeter and use cases are defined in terms of ***Test scenarios.*** One of the prerequisites of a load test design is that the user's population (a reasonable guess) and typical, business critical, frequent, edge cases behaviours are identified, and carefully documented.

With the help of data gathered as a result of load testing, we can decide what, when and how of resources and budget allocation so we can maintain systems to handle low, average, peak and harsh loads in a real production environment.

2. What does it mean by Web Load?

When I say Web Load, I roughly mean to say (from my own web software testing experience), that in today's web/mobile technologies dominated world, the software systems typically include…

- A GUI interface for human users to interact with system's capabilities
- ReST APIs (often on the top of a Microservices architecture)
- Authentication and authorization systems (3rd party)
- Servers including Database, Application, Web, and CDN

- Load Balancers
- Technologies and services provided by Cloud services providers like Azure and AWS.

Now multiple users will end up generating a load on such ***systems*** via ***applications*** (a component of the system) hosted on the web. That's why ***Web Load***. Human users will eventually experience the speed, responsiveness and degradation of such systems via web only.

3. Isn't Load Testing the same as Performance Testing?

This is an interesting question and I would like to answer it this way. To me, as I am learning, Performance; is defined in terms of Users and Stakeholders happiness and measured in terms of ***resources*** e.g., Time, Money, and ***value*** e.g., Repeatability, Satisfaction, and Profit. Load Testing, and other variants (including Front End performance assessment) of performance testing, help testers to gather data about Performance. Sometimes I think this way, and I may be wrong that, If Performance Testing is a ***Class***, the Load Testing can be defined as an ***Object*** of that class.

4. What are those different factors that might badly affect the overall performance of a web application?

Have you heard this proverb saying What you sow so shall you reap? Amongst many 'observable' facts on the implementation side and I will talk about those in the session, one of my findings is that the seed of the performance problem seems to be in the ***people's*** handling of performance requirements within SDLC (Software Development Life Cycle) itself. For example, I have often seen the following...

- Requirements (especially performance requirements) are not primarily and duly discussed, concluded and well documented.
- Testers are not involved from day one on discussions on overall requirements.
- Laziness, ambiguities, and loopholes in the requirements usually cause an inadequate software design and then, poor or shallow design leads to code(s) that produces low performant implementation (not products).
- Less time given for POCs (Proof of Concept) and sometimes MVP (Minimum Viable Product) code moving to Production as it is.

- Development team (including testers) not trained or self-motivated or instructed on adequately thinking of Performance or thinking of UX ***only*** in terms of GUI, Usability and Front-End aesthetics.
- Development team (including testers) not explicitly discussing an under development or under maintenance function or feature's performance aspect.
- Not thinking enough in terms of User's devices, geolocations, use cases, data size, network conditions, server's scaling and so on.

5. ***Is there a connection between User Experience, Non-Functional testing and Functionality of the system?***

I found it very hard to NOT find a connection. Forget about me explaining that there is NO Connection. Thinking of it this way, as a human user, most of the time I am purposefully interacting with software's ***capabilities*** (delivered as a set of features, ***functionalities*** and individual ***functions***) and I desire to ***experience*** the speed, reliability, accuracy, privacy and security. I don't care which terminology of your software jargon's world you want to (mis)use to test the software. Just get it done. So as a tester, why is there a temptation to fall into the trap of explicitly imagining and creating boundaries among UX (User Experience), NFT (Non-Functional Testing) and FT (Functional Testing)? Even if someone says, let's call it Para Functional Testing, I would ask 'Purpose'?

6. ***Why is JMeter a tool of choice for this workshop and not others?***

I prefer this tool because it is Open source, multi-threaded, easy to extract, explore and learn, good documentation, continuous upgrades and relevant deprecations, experimental features, plugins, bug reporting and tracking system, reporting integrations and enhancements, community support, great contributors, StackOverflow, blogs and biasness to name a few of the reasons.

7. ***Can I use JMeter as a load testing tool in my day-to-day testing activities?***

In general, I would say, why not? Yes, as aforementioned, as long as you are curious and thoughtful enough about performance of the SUT (Software Under Test), you can and should use JMeter as a load injector and test tool. At the same time look at your existing tools set, capabilities and organizational policies.

8. ***What are key metrics to measure and report?***

Most of the time, I see people care about Average Response Times and that too in 99 percentiles, but I found (and am still working on to find out) that there is more to that and I will cover that in the workshop. I am still learning ‘Measure what Matters’, so don’t feel bad If I don't give you direct answers here.

9. Are there standards on what an ideal Response Time should be?

Yes and No. Once upon a time I was also under the impression that there is an Ideal Response Time that has to be achieved but later I realized it is about Trade Offs and fault tolerance.

10. Prerequisites

Java installed and JMeter v5.5 extracted on your Windows / Mac machine.
Please refer the attached ***WQF_2023_JMeter_Prerequisites.pdf***