

# NeoLoad: Comprehensive Mobile Performance Testing

Chetan Nimba Aher<sup>1</sup>, Chaitali Subhash Borse<sup>2</sup>

<sup>1</sup>Savitribai Phule Pune University, All India Shri Shivaji Memorial Society's, Institute of Information Technology, Pune-411041

<sup>2</sup>Xpanxion International Pvt Ltd, Sarjaa Road, Sanewadi, Aundh, Pune-411007

**Abstract:** Mobile performance testing is a non-functional test type and there is no performance testing without tool support. The major challenges during the mobile performance testing phase are acquirement of expensive tool licenses and the compatibility of device versions and the testing tool. Neoload is one tool which addresses these issues. This paper is about how Neoload provides solutions for mobile load testing, designing realistic load test, best practices for recording load test scenario, runtime test execution, and result analysis.

**Keywords:** Non-functional Testing, Mobile Load Testing, Quality Assurance

## 1. Introduction

Mobile applications and mobile websites have become a major channel for conducting business, improving employee's efficiency, communicating and reaching consumers. In the past, mobile played a smaller role in business applications, so performance issues and outages were less of a concern. This is no longer the case. Today, performance problems with mobile applications lead directly to revenue loss, brand damage, and diminished employee productivity.

## 2. Challenges encountered by QA team

Mobile application performance is the frightening for QA teams due to the following factors:

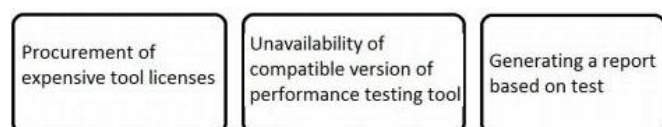


Figure 1: Challenges for QA team

While usage of tools boosts quality and productivity of QA teams significantly, it also requires huge initial investment in purchasing tools and licenses. Investing on license for a non-function tool is what make it more challenging for a QA team. Low rating mobile application can use free edition license but for a big application it is a bit difficult. Another problem is that QA team's faces problem with the compatibility of device versions and performance testing tool. At times the performance tool is unable to recognize certain screen elements used by newer applications and this leads to errors during execution of the test scripts. Somehow QA team manages to run a performance test of mobile devices but there is always a need of results of the test which showcase the performance impacted area the application on devices. If QA team manages to get the result and publish it to client then also its difficult for non-technical person to understand the factors.

To answer all the challenges we can turn to NeoLoad for a better solution in the field of mobile performance testing. NeoLoad Free Edition adds powerful new testing capabilities for mobile applications and web applications to increase testing speed, agility and resulting insights. This solution gives developers and testing professionals the flexibility to run small performance smoke tests with Continuous Integration builds, test performance of new components before integrating to the mainline, and gives testers an opportunity to confirm the configuration of their monitors and protocols.

## 3. NeoLoad Free Edition Benefits

- Load Testing for all Web & Mobile Applications with up to 50 virtual users
- Realistically simulates user activity, analyzes server behavior & pinpoints bottlenecks
- Includes WAN emulation and device simulation out-of-the-box
- Supports the latest Web, Mobile and RIA technologies
- Automated CI server integrations (including Jenkins)
- Fast and Powerful
- Provides Actionable Insight Title and authors

## 4. Mobile Application Types Supported by NeoLoad

To generate a mobile test scenario, it is necessary to identify the application type. Below are the application types supported by NeoLoad.

### 4.1 Web application

Access only through mobile browsers. They are developed specifically to cope with the mobile browsers limitations. They carry URLs just like any other websites. Pure HTML mobile applications are easy to record and to test in Neoload.

## 4.2 Native application

A native application is software written for mobile devices. It is specifically designed to run on a device operating system and machine firmware, and typically needs to be adapted for different devices. Most native applications are designed to rely on an API which uses a proxy. NeoLoad can act as a proxy to catch the activity of the mobile device or emulator. NeoLoad can record test scenarios on a native application when the mobile device supports the definition of a proxy — by default, IOS and Android 4.0 (Ice Cream Sandwich) and when the application is able to use the proxy set on the device.

## 4.3 Hybrid application

A hybrid application is a web application packaged as a native application. The native part of a hybrid application is limited to a few features like a menu or the automatic login. Whereas the web main content is displayed in an embedded web browser component. The standardized content of hybrid applications makes it possible to reuse it on a number of platforms. Hybrid applications are distributed through the platforms application stores. When testing hybrid application, the application should be processed the same way as native applications.

## 4.4 Secured application

When the mobile application records uses a secured connection (HTTPS) to connect to the server, the mobile device can see the NeoLoad recording proxy or the NeoLoad tunnel as a man-in-the-middle attack and refuses the connection. It is necessary to authorize the connection with the proxy or the tunnel to be able to record the traffic. The NeoLoad root certificate must be imported in the device or emulator. The root certificate is created when NeoLoad is first launched and is named NeoLoad\_Root\_CA.cer.

## 5. Monitoring Modes in NeoLoad

**a)Proxy mode:** The communication between the device or emulator and the server is recorded through the proxy based recorder. Both NeoLoad and the mobile device must be on the same network. The mobile device must be connected to the NeoLoad network in Wifi. On the mobile device, it is necessary to manually modify the proxy used by the application or by the browser. The hostname must be the name or the IP of the NeoLoad machine. The port is the NeoLoad recorder port (8090 by default).

**b)Tunnel Mode:** This monitoring mode is a second way to record an application with NeoLoad. It is used when the proxy recording is not possible. This requires some prerequisites like administrator rights on Neoload machine. For recording with tunnel mode Neoload will act as DNS server for mobile, so modify mobile DNS settings to use Neoload machine as DNS server.

**Note:** NeoLoad can record mobile applications in many ways: directly from the device itself, from an emulator, or from a desktop web browser for pure HTML applications.

## 6. Neoload tool features

### 1) Design Tab: Allows user to create virtual user scripts.

- a) **Virtual user:** All the scenarios and the scripts are added and displayed
- b) **Population:** Here we can define number of users going to use the app as same time.

### 2) Runtime Tab: Allows user to run created Virtual scripts and the Load test.

### 3) Results Tab: This will display result of the Virtual user scripts and load test result and analysis. This includes the analysis graphs generated by the test.

We will see brief of all these tab in next section with all other functions and parameters provided by NeoLoad for betterment of mobile performance testing of the mobile application.

## 7. Mobile Load Testing flow with NeoLoad

- Recording Http traffic between browser and server.
- Defining Test Parameter/Scenarios
- Running the Test
- Analyzing the result.

Once Mobile device is connected by the Neoload then tool can record an HTTP application as a go-between with the browser and the web server. With a secure connection, NeoLoad has to decipher the traffic flowing between the two entities in order to analyze its content. Each time a communication is sent, NeoLoad decodes the incoming data, analyzes it, and re-ciphers it before sending it out again. This process requires the use of certificates. NeoLoad records all the HTTP flows going through the machine. It is necessary to close HTTP clients or define exclusion rules in NeoLoad to avoid unwanted requests in the recording. Starting a new recording of test scenario creates a new User Path and places all the web pages visited during the recording into that User Path. Before starting the recording check mobile device is added to the Client settings and some advanced settings like Protocols, Modes and support settings of the User Path. The User Path is a succession of web pages, which may contain logical actions such as Containers, loops or delays to create a more complex behavior. In NeoLoad, User Paths are created and managed in the Design section. In this section also deals with Logical action, Frameworks, Population and Monitors. The Runtime section displays real-time information while the test is running. Under which a scenario contains all the settings required to run a load test. Then test duration of the population, Population load policy, Zones and load generators running the population and advanced settings for populations and scenarios can be configured for each of the selected Populations. NeoLoad provides two test modes, normal and debug. In normal mode, the test starts with all the System resources used to generate the Virtual Users. The debug mode allows validating the various User Paths defined in the scenario. The Runtime sections also support the runtime overview, runtime graph, runtime errors, runtime alerts and runtime users of test scenario. Where Runtime

overview monitors test scenario in the form of Graph, controller and Load generator. The graph display main statistics are plotted in real time: number of Virtual Users currently running, error rate (errors/sec.), number of alerts, hits/sec. and average request response time. Under Controller two metrics monitor the Controller activity: Controller CPU Load and Controller Memory used. Load Generator monitors the Load Generators activity: Hit rate, errors, Virtual Users, CPU Load, Memory used and throughput. Statistics, monitors set on the servers and External data can be plotted on graphs during the test runtime. When the test is first run, NeoLoad creates default graphs. These can be modified in Runtime Graphs during the test, or new graphs can be created. When subsequent tests are run, NeoLoad restores the settings of the graphs used in the previous test. In Runtime Error sub section, error details like time, duration, size, load generator, URL, virtual user, request server response, assertion etc. can be obtained while the test is running. Alert details like time, severity, name, graphs can be obtained while the test is running in Runtime Alerts. The Runtime Users tab allows monitoring the Populations and Virtual Users runtimes in real time and to changing the test conditions while the scenario is being played out. The Results section includes all the measures computed after a load test. The Result section have Test summary, Values, Graphs, Errors, Alerts, Logs and Details tab. In every tab of the section, the Results drop-down list makes it possible to select which test results to view within the current project. The selection is reflected in all subsequent result panels. The Test Summary tab contains all the main statistics for the test like NeoLoad Test Report and main graphs. The Tools button allows generating a report for the selected test, among other functions. Values tab allows sorting and plotting elements of a test quickly. Graphs tab allow to visualize how statistics evolve over the test duration. Errors tab contains details of the HTTP request errors. Alerts tab displays details of the alerts that were triggered on performance counters during the test. Logs tabs allow to view the log files of the various Load Generators and Monitoring Agents used during a test. The Details tab is enabled when the scenario has been launched in Debug mode or when the project contains external data. NeoLoad allow generating reports based on test results containing predefined data, graphs and statistics. A report may also contain custom graphs and comments. A report can be generated from the results manager, summary panel or command line. NeoLoad allow creating reports with customized parameters report, comparison report and also filtered test report of test result. When NeoLoad is used in Collaborative mode, results of test can be easily shared with the members of the project teams. Publishing a test result is necessary to make it possible of the other test team members to analyze it. A test result is generated locally after a test is performed. In order to share it, it must be pushed onto the collaboration server from which the other testers can retrieve it onto their local machines.

## 8. Heavy load testing in NeoLoad

NeoLoad makes it possible to build quality Web applications that offer high performance and scalability and that are far more likely to be used and appreciated than applications that

are buggy and slow. However, building even the best software still requires that the application be tested under heavy loads (regardless of the specific goals for the application—production tests or stress tests). There are three average levels in heavy load tests starting from 1,000 Virtual Users to 30,000 and above Virtual users. Where 1,000 to 10,000 virtual users are tested on premises, 10,000 to 30,000 Virtual Users are tested with Cloud Load Generators and 30,000 and above Virtual Users are tested in Full Cloud.

## 9. Benefits of NeoLoad over other Tool

When compared top tools for mobile performance testing then it has been seen that NeoLoad is way better than other tools. Cloud and mobile testing is not supported by Jmeter whereas it is supported by Neoload. Comparison of the results with previously run results was something that was missing from Jmeter. Neoload has an option for this. The result summary provided by Jmeter is very minimal as compared to Neoload. A different type of graphs available to the tester is more in Neoload than in Jmeter. NeoLoad is about 10% of the cost of Load Runner, and Neoload includes technical support as well as perpetual licensing. This factor shows that NeoLoad has an edge over other tool for mobile performance testing.

## 10. Conclusion

Mobile performance testing is a non-functional test type which should not be underestimated. New methodologies and technologies provide increasing opportunities, but also more complexity for mobile application systems. Therefore this paper is all about NeoLoad as solutions to most of the challenges faced by the QA teams in terms of procurement of expensive tool licenses, compatibility of device versions and the testing tool and reports generated based on the load test results. This paper also gives a brief view of NeoLoad features, functions and benefits over other tools

## References

- [1] <https://www.infosys.com/IT-services/validation-solutions/white-papers/Documents/performance-testing-functional-test-tools.pdf>
- [2] <https://www.neotys.com/resources/mobile-load-testing-challenges>
- [3] Neoload document :
- [4] <http://www.neotys.com/documents/doc/neoload/latest/en/html/#5660.htm>
- [5] [4] [https://www.itcentralstation.com/product\\_reviews/neotys-neoload-review-30434-by-nikolai-grabner](https://www.itcentralstation.com/product_reviews/neotys-neoload-review-30434-by-nikolai-grabner)