White Paper

Mastering Peak Traffic with Performance Testing

Best Practices with Performance Testing Portfolio

opentext™

Table of Contents

page

From insurance to entertainment, from government to higher education, organizations have to prepare their websites and apps for massive volumes of concurrent users.

Are Your Online Services Ready for Peak Traffic?

Most people would say that Mother's Day, the Super Bowl, and the election of a Pope have absolutely nothing in common. But on closer inspection, all three events produce a similar result: They generate significant online traffic.

Contrary to popular perception, occasional spikes in demand for online services aren't just a concern for retailers. The same story plays out across virtually all industries. From insurance to entertainment, from government to higher education, organizations have to prepare their websites and apps for massive volumes of concurrent users. Sometimes they can see the deluge coming in the form of predictable seasonal spikes in demand, but often they can't.

This is where the performance testing conversation comes into play. Most people would look at a massive crash like that and say that it could have been prevented if they had only tested and planned properly. That sentiment is true—as long as they knew what they were testing.

8 Best Practices for Handling Peak Traffic: a Complete Approach

When peak traffic hits, both external applications and internal applications are impacted, causing losses in revenue, brand reputation, and employee productivity. To make sure you are ready for a peak season—whenever it hits—here are eight best practices to implement an effective performance test for both consumer-facing and internal applications.

1. Identify Performance Targets (or Goals)

This step is critical to interpreting your testing needs and is used to determine whether the system can scale and perform to your specifications. At this stage, you need to translate your user requirements into performance testing objectives (such as response time for each transaction). A thorough evaluation of the requirements before beginning load testing can help provide realistic test goals and conditions. A crucial Key Performance Indicator (KPI) for the peak season is how much traffic goes to the company's website and/or internal application. For website traffic, you can use a free service available on the Internet that estimates daily page views/daily visitors of a website (for example, siteworthtraffic.com). For internal traffic, you can use information provided by the production environment (such as logs).

2. Script and Emulate Business Processes

A virtual user (Vuser) emulates the real user by interacting with the application as a client. You must identify and record all the various business processes from start to finish.

The right preparation and tools can help to make sure that organizations don't suffer catastrophic scalability or stability issues under high-traffic situations.

CREATING A VIRTUAL USER SCRIPT WITH OPENTEXT VIRTUAL USER GENERATOR (VUGEN)

When testing an environment, you need to emulate the true behavior of users on your system. OpenText™ testing tools emulate an environment in which users concurrently work on, or access, your system. To perform this emulation, the human is replaced with a Vuser. The actions that a Vuser performs are typically recorded in a Vuser script. The primary tool for creating these scripts is OpenText™ Virtual User Generator, also known as VuGen. You can use VuGen scripts across all the OpenText™ Performance Testing Suite (OpenText™ LoadRunner Professional, OpenText™ LoadRunner Enterprise and OpenText™ LoadRunner Cloud*).

The following table lists the supported Vuser protocols.

.NET	Java Record Replay	POP3	SMTP (Simple Mail Protocol)
C Vuser	Java Vuser	RDP (Remote Desktop Protocol)	Teradici PCoIP Protocol
Citrix ICA	JMeter	RTE (Remote Terminal Emulator)	TruClient—Mobile Web
DevWeb	LDAP (Listing Directory Service)	SAP GUI	TruClient—Native Mobile
(DNS) Domain Name Resolution	MAPI (Microsoft Exchange)	SAP—Web	TruClient—Web
FTP (File Transfer Protocol)	ODBC	Selenium	Web—HTTP/HTML
Gatling	Oracle—2 Tier	Siebel—Web	Web Services
IMAP (Internet Messaging)	Oracle—Web	MQTT	Windows Sockets
Java over HTTP	Oracle NCA	SMP (SAP Mobile Platform)	

TRUCLIENT SCRIPTING

Testing modern applications effectively will ensure you have time to handle issues, not simply identify them. For that, we created DevWeb. Your application developers can create simple load testing assets, focused on network activity to load and measure performance. These tests are created and executed with their favorite IDE, using an intuitive JavaScript SDK. Combine the DevWeb script along with a TruClient script to have an effective way to capture application behavior as your users would experience it.

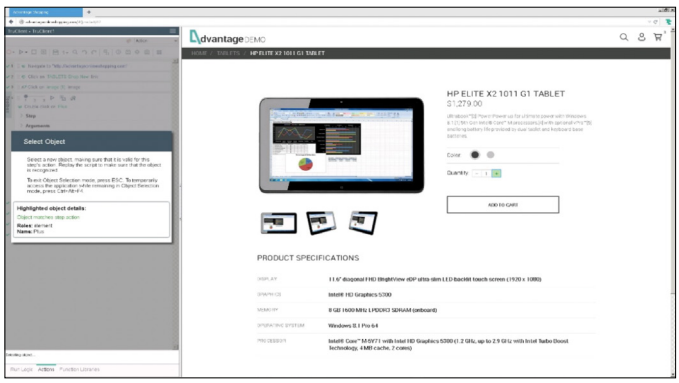

Figure 1. TruClient scripting

** with some limitations*

TIP: Peak traffic rates are usually 2–3 times the usual volume.

SCRIPT REUSE

LoadRunner Cloud supports popular open source testing tools such as JMeter and Gatling, so DevOps teams can easily upload existing scripts created with these tools into LoadRunner Cloud and run them. LoadRunner Cloud and LoadRunner Enterprise enable you to run JMeter scripts and integrate JMeter with additional script types in any performance tests.

LoadRunner Cloud supports also integration with widely adopted open-source tool, Selenium. Selenium scripts are very similar to load testing scripts, that is they simulate user steps, keyboard inputs, behavior in different browsers, and so on. By reusing the code written for the functional test and modifying it in the context of performance testing, while being careful that it applies to the same use case, DevOps teams obviously save time. In addition to the timesaving, they also experience better test coverage and can rest assured that the user experience has not been compromised.

OpenText™ TruClient is an innovative, browser-based virtual user generator that supports simple Web as well as modern JavaScript-based applications.

3. Plan and Design Tests

You should identify key scenarios, determine variability among representative users and how to simulate that variability, define test data, and establish metrics to be collected. Consolidate this information into one or more models of system usage to be implemented, executed, and analyzed.

Performance tests are usually described as belonging to one of the following four categories:

- **Performance Test:** To determine or validate speed, scalability, and/or stability
- **Load Test:** To verify application behaviour under normal and peak load conditions
- **Stress Test:** To determine or validate an application's behaviour when it is pushed beyond normal or peak load conditions
- **Capacity Test:** To determine how many users and/or transactions a given system will support and still meet performance goals

Each type of test can help identify and address the risks related to peak traffic:

Test Type	Risk(s) Addressed
Capacity	■ Is system capacity meeting business volume under both normal and peak load conditions?
Volume	■ Will performance be consistent over time? ■ Are these slowly growing problems that have not yet been detected? ■ Is there external interference that was not accounted for?
Load	■ How many users can the application handle before undesirable behavior occurs when the application is subjected to a particular workload? ■ How much data can my database/file server handle? ■ Are the network components adequate?
Spike	■ What happens if the production load exceeds to the anticipated peak load? ■ What kinds of failures should we plan for? ■ What indicators should we look for?
Stress	■ What happens if the production load exceeds to the anticipated load? ■ What kinds of failures should we plan for? ■ What indicators should we look for in order to intervene prior to failure?

PLAN FOR DIVERSE NETWORK CONDITIONS

App performance and the user experience are affected by network latency. Excessive network latency or packet loss doesn't just make your app respond slower, it can cause the app to behave erratically or even fail outright. One of the keys to addressing this challenge is to use OpenText™ Network Virtualization (NV) to simulate realistic network conditions in your load tests. Network Virtualization (NV) helps you accurately test the impact of the network on apps long before they are put into production. Once you understand the impact of the network, take the next step: optimize and fine-tune your app to make sure that it will perform well in the expected network conditions. Network Virtualization is embedded in all LoadRunner solutions.

Four Major Categories of Performance Tests:

- Performance Test
- Load Test
- Stress Test
- Capacity Test

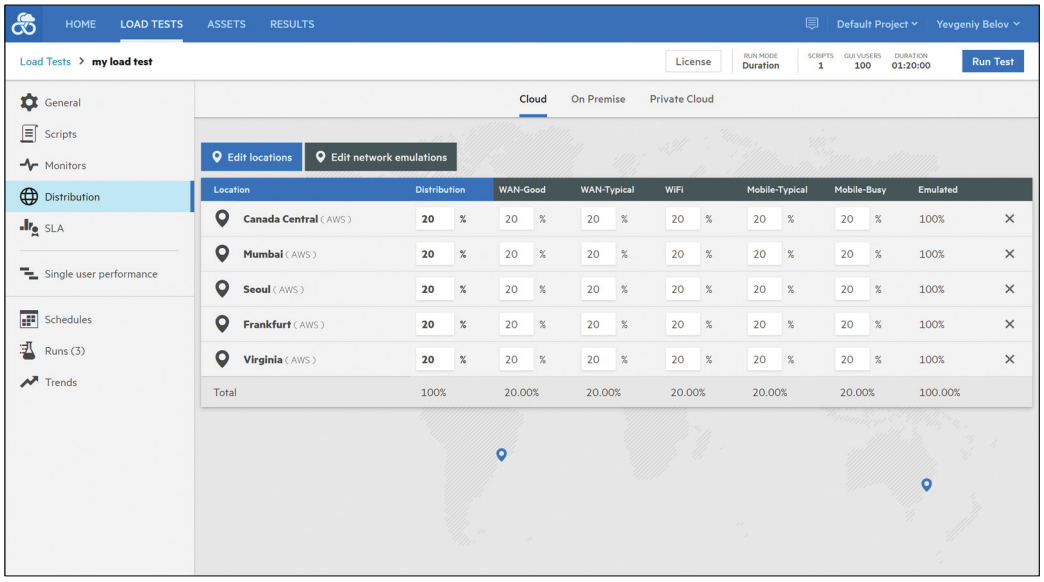

Figure 2. Edit locations in LoadRunner Cloud

4. Execute the Test

Among those new to performance testing, there is often a misconception that execution is a single event. In fact, it is a multi-step process consisting of several types of performance tests.

Network Virtualization is embedded in all three LoadRunner solutions at no additional charge.

Creating a baseline is the process of running a set of tests to capture performance metrics for the purpose of evaluating the effectiveness of subsequent performance-improving changes to the system or application. After executing the baseline test, you should execute the following ones:

With LoadRunner Cloud you can scale easily for more than 5 million Vusers without requiring hardware; it does everything automatically and quickly via the cloud.

Debug Run

- Run 5–10 virtual users
- Eliminate data concurrency errors and fundamental issues
- Use appropriate think time and full extended logging

Isolate Top Time Run

- Run 20% of full load
- Isolate transactions that are taking an unusually long time
- Use appropriate think time and standard logging

Smoke Test

- Run 100% of full load for short time
- Do not report the results as official or formal parts of your testing
- Use appropriate think time and standard logging

Full Load

- Run 100% of full load
- Perform actual test run comparing results with test goals
- Use appropriate think time and just-in-time logging

SCALE

We have to be honest: creating a load in terms of calling an API with a set number of requests/sec for a few virtual users is possible, but it can give you the wrong results. Aspects of user behavior, such as think time, need to be taken into account. So we need to execute the expected load with the actual number of potential users. How many users? 1,000? 100,000? 1 million?

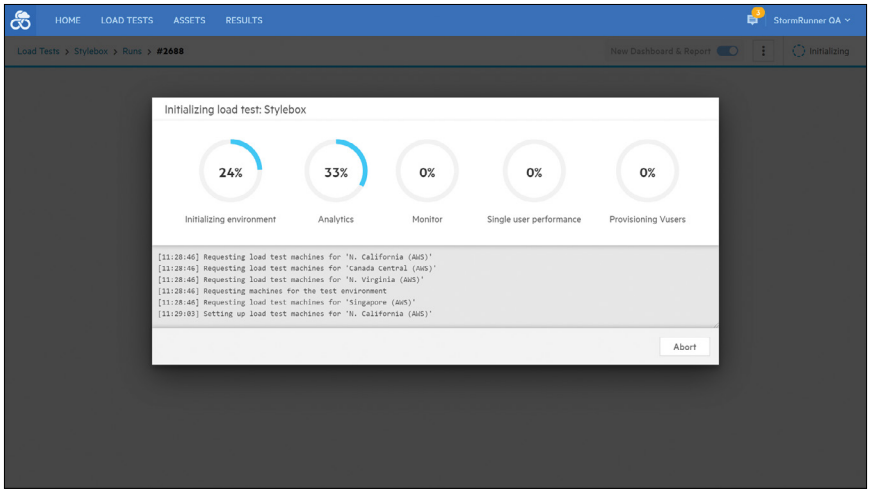

Figure 3. LoadRunner Cloud scales for millions of Vusers

With LoadRunner Cloud you can scale easily for more than 5 (five) million Vusers without requiring hardware; it does everything automatically and quickly via the cloud. There is no need to spend time and money setting up hundreds of load generators.

CONSIDER PERFORMANCE TESTING FROM THE CLOUD

Your customers are coming from all over the world, so to gain a realistic view of performance you need to test from all over. Because of this, cloud-based load testing is essential for testing performance against spikes in demand. Using cloud load-testing generators is the only way to scale quickly and add users from different worldwide e-locations.

LoadRunner Cloud is a cloud-based performance testing solution: you can spin up load generators via multiple cloud vendors such as Amazon Web Services (AWS), Google Cloud Platform (GCP), and Azure, and you can see the results of your tests and gain real-time insights into performance issues instantly. If you are using LoadRunner Professional and LoadRunner Enterprise and you need additional firepower, you can easily add cloud-based load generators to a scenario.

If you are using LoadRunner Professional and LoadRunner Enterprise and you need additional firepower, you can easily add cloud-based load generators to a scenario.

5. Monitor Performance and Availability

Monitoring the performance and availability of your software applications is an important best practice, and Application Performance Management (APM) tools are essential.

APM TOOL INTEGRATION

APM tools are used to monitoring and manage of performance and availability of software applications by reporting mainly the following metrics:

- End user experience (for example, response time)
- Server performance (for example, CPU, memory, server response time, code hotspots, exceptions)

So your tool will generate load, while the APM tools will mostly monitor server behavior. Using them in conjunction, teams can better identify performance drawbacks before real users find them.

LoadRunner Professional, LoadRunner Enterprise and LoadRunner Cloud support Dynatrace and New Relic enabling you to:

- Integrate Dynatrace/New Relic graphs during online execution
- Measure hundreds of metrics in each test run
- View combined results in analysis report

ONLINE ANOMALY DETECTION WITH LOADRUNNER ENTERPRISE

The anomaly detection capability of LoadRunner Enterprise enables teams to use powerful analytics to visualize and spot anomalies and performance problems and find root causes using real-time metrics. Engineers can use these insights to speed their diagnosis and investigation into system performance, and even see the precise triggers that caused the anomalies.

3 Phases to Pinpointing Bottlenecks During Results Analysis:

- Compare results against goals
- Identify potential bottlenecks
- Correlate results

Figure 4. Online anomaly detection

6. Analyze Results

Performing root cause analysis means drilling down from the most general reports to localized metrics. The combination of complex applications and dynamic characteristics of network traffic can significantly degrade application performance. Performance problems can occur at many points along the route between the application and users.

There are typically three phases to pinpointing bottlenecks during results analysis:

- **Compare results against goals:** confirm when performance has not met expectations.
- **Identify potential bottlenecks:** list all the pieces of the system that might have caused the slowdown.
- **Correlate results:** determine the most likely cause by correlating transaction times and backend monitor metrics.

FOCUS ON THE END-USER EXPERIENCE

In your performance testing, focus on load times and intuitive flows. And keep the user's perspective in mind. There are many ways to improve perceived load times compared to actual load times, such as optimizing image sizes and rendering certain content to appear first, such as the top of a page.

The NV Insights Report is a comprehensive network analysis report that provides information about how your application performs over various networks, during a scenario run. The resulting NV Insights Report can help to pinpoint root causes for performance issues, and provide optimization recommendations to resolve the issues, thereby improving the performance of your application.

With the Client-side Breakdown Report, your teams can view statistics that help you measure the quality of the user experience on your application.

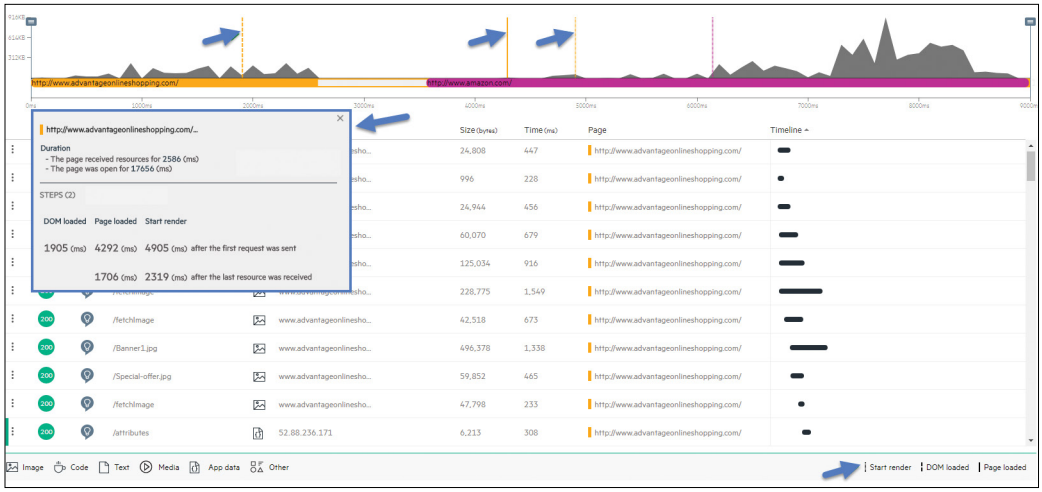

Figure 5. Client-side Breakdown Report

Network Virtualization receives the time of the events (Start render, DOM loaded, Page loaded), allowing the user to inspect events in the client as they correlate to network calls.

ANALYSIS WITH LOADRUNNER PROFESSIONAL AND LOADRUNNER ENTERPRISE

Our testing and operational environments are becoming increasingly complicated. Tools can help us to step past the complexity and perform root cause analysis quicker. The Analysis tool—a unique tool in the performance testing space—provides graphs and reports enabling you to view and understand the data, and analyze system performance after a test run, helping you to find bottlenecks and thus improve the performance of your application.

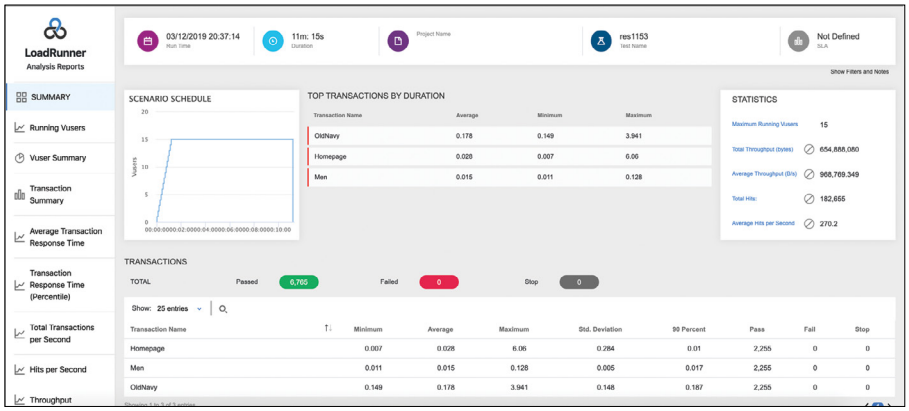

Figure 6. LoadRunner Professional Analysis Summary

Things to Look Out for During Execution:

- The sudden appearance of errors
- A sudden drop in transaction throughput
- An ongoing reduction in available server memory

The LoadRunner Enterprise trend reports allow you to compare performance test run data over time, giving you better visibility and control of your application's performance. By comparing the same measurement in more than one instance of a test run, you can identify whether its performance trend is improving or regressing.

7. Analyze the Trend Report (and Retest)

The correct interpretation of results is obviously vitally important. Assuming that you've (hopefully) set proper performance targets as part of your testing requirements, you should be able to spot problems quickly during the test or as part of the analysis process at test completion. What's important is having all the necessary information at hand to diagnose when things go wrong and what happened when they do.

TREND REPORT IN LOADRUNNER ENTERPRISE

The LoadRunner Enterprise trend reports allow you to compare performance test run data over time, giving you better visibility and control of your application's performance. By comparing the same measurement in more than one instance of a test run, you can identify whether its performance trend is improving or regressing. Trending information can now be viewed directly within Jenkins, giving you the metrics you need sooner without logging onto LoadRunner Enterprise.

Figure 7. LoadRunner Enterprise Trend Report and Jenkins

8. Establish Continuous Performance Testing

Without question, continuous performance testing is critical to the success of your apps. If you don't test until the end of the cycle, you risk production issues, user complaints, poor reviews, and damage to your brand. So in Agile development/DevOps environments, performance testing must be integrated with the whole development process.

LoadRunner Cloud integrates with your open-source CI/CD frameworks such as Jenkins, Azure DevOps and Bamboo. This integration lets you fire up performance tests to check that new code you've pushed out hasn't caused any regressions or broken your compliance with performance SLAs.

Connect with Us
OpenText CEO Mark Barrenechea's blog

LoadRunner Cloud also integrates with AWS CodePipeline. This powerful utility helps you automatically build, test, and deploy your applications in the Amazon Web Services (AWS) cloud. LoadRunner Cloud relies on a public REST API that can be used to trigger tests and collect results from CI/CD tools, and the command line interface (CLI) tool provides a less technical way to interact with LoadRunner Cloud. If you're running Linux, for example, you can simply use the CLI tool to run a specific test without the need to open a browser and perform multiple steps. LoadRunner Cloud integrates with the open-source Git version control system, enabling you to upload scripts from your Git repository.

LoadRunner Professional and LoadRunner Enterprise continue to introduce new features and integrations to incorporate load and performance testing into continuous integration and continuous testing practices. LoadRunner Enterprise features strengthened integration with Jenkins, and now includes Trending *(see above)*, while LoadRunner Professional supports testing in Agile and DevOps environments as well (jUnit, nUnit, Eclipse, MS Studio, Selenium, Jenkins, Git).

LoadRunner Professional and LoadRunner Enterprise continue to introduce new features and integrations to incorporate load and performance testing into continuous integration and continuous testing practices.

Innovations We Are Delivering

The Performance Testing suite empowers teams to manage testing and performance complexity. LoadRunner Cloud, LoadRunner Professional, LoadRunner Enterprise, and Network Virtualization work together to enable teams to thrive in today's chaos and deliver predictable quality across all development environments.

Learn more at
www.microfocus.com/performancetest
www.microfocus.com/opentext

opentext™