Modernize Application Performance Testing
# Table of Contents

- The Age of Complexity ......................................................... 1
- Breathe New Life into Application Performance Testing ............... 1
- Continuous Performance Testing: How to Make It Happen ............ 2
- Do You do DevOps? ................................................................. 4
- Testing New Technologies ..................................................... 6
- Rely On Open Source and Collaboration .................................. 8
- Use Tools That Reduce Complexity ....................................... 9
- Why You Need Software Lifecycle Virtualization ....................... 11
- Find the Right Application Testing Platform ............................ 14
The Age of Complexity

Software evolution is a fact of life. Application architectures and platforms are rapidly evolving. Along with web and mobile, software is becoming fully meshed into the fabric of daily life through the Internet of Things (IoT). This evolution has powered previously unthinkable innovations: for example, cell phones and wearables are now interconnected, and you can monitor your doorbell from a beach thousands of miles away.

We are seeing a seismic shift in how software is delivered. The increasing complexity from new software architectures and new platforms is simply too much for delivery teams to manage. Systems are increasingly software-reliant and interconnected—making design, analysis, and evaluation harder than ever before.

Delivery teams today wrestle with apps built with:
- Shared services and APIs
- Widespread use of open-source code and tools
- Different protocols for IoT and mobile application delivery

These teams need smarter testing solutions that cut through the noise. Typical testing labs simply can’t keep pace with the proliferation of application and device types. With multiple browser variations and the growing diversity of mobile operating systems (and versions), plus the anticipated surge in IoT, no tester or physical test lab can keep up with ever-increasing user expectations in the face of accelerating release cycles, mushrooming IT complexity, and massive scalability requirements. Throw in new initiatives such as digital transformation, the transition to Agile methods, incorporation of Big Data and predictive analytics, and exploration of the Internet of Things (IoT)—and complexity increases exponentially.

Breathe New Life into Application Performance Testing

We believe that an application-performance testing platform ready for this challenge needs the following capabilities:
- Easy integration with multiple tools/technologies and support for open APIs: Your testing platform should integrate and enhance the value of widely used open-source development, testing, and DevOps tools, making it easy to combine with continuous integration (CI) and continuous deployment (CD) frameworks such as Jenkins. Developers should also be able to use open source scripting tools such as JMeter, and the Git version control system for managing scripts.
A plan for handling diverse network conditions: Application performance and the user experience are affected by network latency. Excessive network latency and/or packet loss don’t just make your app respond slower; they can cause the app to behave erratically or even fail outright. Your testing platform should be able to handle complex and demanding network conditions.

Focus on the end-user experience: Your performance testing platform should help you focus on load times and intuitive flows, while keeping 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. These sorts of optimizations can make it appear that a site is loading quickly.

Ability to use production analytics: Testers always have more tests than time. Production analytics solve this by providing views into how apps are actually being used in the wild to focus testing. Your testing platform should enable you to use production analytics to gain insights into the real usage of your applications.

This white paper explores the above challenges and analyzes how testers must change their methodologies to keep pace with technology and user expectations. It also provides a detailed overview of the latest features of Micro Focus® LoadRunner and Performance Center software—the industry-leading performance validation tools.

Continuous Performance Testing: How to Make It Happen

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/DevOps environments, performance testing must be integrated with the whole development process.
Continuous testing means running tests at each stage in the pipeline, giving the team feedback to improve quality and increase their velocity. Tests are triggered automatically by events such as code check-in. The faster defects are found, the faster they can be addressed. Continuous testing doesn't happen by itself. Instead, it requires a significant shift for development teams to succeed.

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

**Unit Tests Execution with LoadRunner**

When executing a performance-testing scenario as part of continuous integration and continuous testing process, the most important thing is to get a clear “pass/fail” indication of whether the application’s performance has been hit as a result of recent changes in the code. You need this information without doing any data processing, just referring to the service-level agreements (SLAs).

In a few simple steps, you can protect your application from performance regressions:

- **Create Vuser Scripts or LoadRunner Tests in Visual Studio or Eclipse:**
  To support deep and effective integration of unit or functional tests, LoadRunner provides add-ins for Microsoft Visual Studio and Eclipse.

- **Add LoadRunner APIs to unit tests in Visual Studio or Eclipse:**
  Extending your test with all the power of LoadRunner’s infrastructure is as easy as clicking on “Add LoadRunner API reference” in the “DevOps Vuser” menu. This is created when you install the add-in. After this simple action, you can use LoadRunner functionality such as transactions, messaging, think time, and so on in your code.

- **Execute unit tests using the LoadRunner Engine:**
  To support the execution of unit tests as part of LoadRunner load test scenarios, the LoadRunner Controller has the ability to open not only LoadRunner scripts but also system or unit tests. You can continue with scenario creation just as you did before with regular LoadRunner scripts.

**Jenkins Integration**

The Application Automation Tools plugin for the Jenkins continuous integration server provides a mechanism for executing LoadRunner Controller scenarios as part of a build script. This plugin allows you to trigger a test as a build step and present the results in the Jenkins’s user interface. You can only integrate scenarios that have service level agreements (SLAs). This allows you to quickly determine whether the test passed or failed and if performance was affected.

Performance Center also features strengthened integration with Jenkins. For example, trending information (which allows you to compare performance test run data over time) can now be viewed directly within Jenkins, giving you the metrics you need sooner, without logging on to Performance Center.
DevOps attempts to remove artificial impediments to software delivery and empowers the delivery team to be responsible and accountable for the software they build and ship.

Do You do DevOps?

Today’s organizations are increasingly adopting a DevOps approach, bringing the development team and the operations team closer together to build high-quality software that can be released to production at any time. This approach succeeds only by keeping the source code free of defects—detecting problems as soon as they are introduced so that they can be weeded out before they get close to the customer. DevOps attempts to remove artificial impediments to software delivery and empowers the delivery team to be responsible and accountable for the software they build and ship.

Performance Application Lifecycle (PAL)

As performance engineers, we need to design a scenario that simulates real-user loads. So why don’t we use information provided by the production environment? Feedback and measurement from production is a good way to improve continuous assessment. For example, real users never use many features and functions, so maybe you don’t need to design a test to cover it. In most cases, developers don’t have the end-to-end visibility that they need, and often they turn to IT admins in search of relevant metrics and data—but the result is generally inadequate.

Performance Application Lifecycle (PAL) in Performance Center enables you to share performance information between development and operations to better plan your performance tests and application deployment. PAL allows for complete end-to-end testing and DevOps feedback.

Users can accurately compare performance test results with real production data benchmarks. Analysis of these results provides a framework to assist you in creating performance test scenarios that resemble realistic environments as closely as possible. PAL enables you to use user traffic and system monitoring data in production to design a performance test that closely resembles production behavior. You can import production data from Microsoft IIS W3C Extended Log Format (IIS W3C), Apache, and Real User Monitor (RUM).
The PAL flow includes the following main steps:

1. **Import**: Import a data set from a production system. Different production monitoring systems provide different data that may influence which information is available to the user.

2. **Create**: After uploading your data set to Performance Center, it analyzes the data and creates a PAL scenario with business flows. You can translate each business flow into a script.

3. **Run**: Run your performance test.

4. **Compare**: Compare performance test results with your production data. If necessary, re-adjust and re-run your test.

**Application Performance Management (APM) Integration**

When applications go into production, the path to high performance on an ongoing basis lies in the aggressive monitoring of the applications and the end-user experience, followed by the process of feeding performance data back to your development teams. This proactive work is one of the keys to avoiding emerging performance issues with production applications. Throughout the monitoring process, the goal is to fully understand the end-user experience and rapidly diagnose any performance issues. To meet this goal, your monitoring toolset should include capabilities for monitoring the real-user experience, so you know what’s actually happening in the world of end users.

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

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

LoadRunner and/or Performance Center will generate load, while the APM tools will mostly monitor server behavior. By using them in conjunction, teams can better identify performance drawbacks before actual users find them. The adoption of APM tools provides DevOps benefits, such as:

- Fast feedback between development and operations
- Increased application resilience from early, proactive monitoring
- Lower operations costs
- Decreased time firefighting production issues

---

**Figure 3. PAL Steps**

1. **Import**: Import a data set from a production system. Different production monitoring systems provide different data that may influence which information is available to the user.

2. **Create**: After uploading your data set to Performance Center, it analyzes the data and creates a PAL scenario with business flows. You can translate each business flow into a script.

3. **Run**: Run your performance test.

4. **Compare**: Compare performance test results with your production data. If necessary, re-adjust and re-run your test.
Performance Center REST API

Relying on an open API-driven integration layer is crucial for a DevOps team because it integrates and enhances the value of widely used open-source development, test and DevOps tools, through integrated data, task and workflow. When it comes to designing web APIs, no other style is more popular than REST.

REST (REpresentational State Transfer) has become the standard de-facto for designing powerful APIs that run over HTTP. The Performance Center REST API enables you to write applications to create and run load tests without using the Performance Center user interface. Performance Center 12.55 includes REST API functionality for:

- Get list of runs—cross projects
- Delete runs—cross projects
- Get runs statistics
- Define SLA
- And many more

Testing New Technologies

How do you make continuous performance and load testing practical in an age of accelerating development cycles and ever-increasing user expectations? For application teams, the mentality must change from simple “record/playback” testing that occurs late in the product cycle to a more robust engineering approach that starts early in the cycle and occurs continuously.

Protocol-Level Scripting

New technologies are emerging daily and are changing the way people and business customers consume content, so we continue working to align protocols to these latest technologies. LoadRunner and Performance Center software now support HTTP/2, the latest version of the HTTP network protocol used by the web. The goal of designing HTTP/2 was to create a faster network protocol while preserving the HTTP methods format. Tests show that web pages are loaded two to four times faster when they are served with HTTP/2.

Since websites are offering more media streaming than ever before, LoadRunner and Performance Center bring full support for audio and video streaming testing (HTML5 and HLS). This enables new capabilities for recording and replaying web HTTP/HTML and mobile scripts as well as new streaming metrics to help identify potential performance bottlenecks.

Because LoadRunner and Performance Center support more than 50+ technologies and platform, you can easily test a huge range of applications including web, mobile, Ajax, Flex, HTML5, .NET, Java, GWT, Silverlight, SOAP, Citrix, ERP and more. LoadRunner supports the latest application technologies as well as legacy ones.
UI-Level Scripting

The complexity of new technologies, the lack of commonly recognized and accepted standards, and the sheer multitude of emerging frameworks and toolkits make it difficult for companies to build Web 2.0 testing strategies and select appropriate automation solutions. TruClient, introduced in version 11.00, is an innovative, browser-based virtual user generator (VuGen) that supports simple Web as well as modern JavaScript-based applications. TruClient Coded (technical preview) is the new functionality introduced in version 12.55 to convert TruClient scripts in code. It makes it easy to code load testing into your mobile and web apps, so testing can occur earlier, more often, and with much less effort over the entire development lifecycle.

Benefits:
- Ability to create complex scripts
- Faster-processing user interface
- Flexibility: DevOps and Agile teams can test their way, using their familiar tools, on their schedule
- Less memory consumption

Figure 4. TruClient Coded
Innovations We’re Delivering: IoT Ready

The Internet of Things is an ecosystem where objects are connected together to exchange data. The system’s complexity doesn’t matter. The important aspect is that an object is identified by its capability to connect to the network and exchange information.

MQTT (MQ Telemetry Transport) protocol is a Machine to Machine (M2M) message-based protocol widely used in Internet of things. It is extremely lightweight and for this reason, it is adopted across the IoT ecosystem. The protocol uses a publish-subscriber paradigm in contrast to HTTP based on request/response paradigm. It uses binary messages to exchange information with a low overhead.

For these reasons, we’ve implemented a new MQTT protocol in LR/PC 12.55. When using this new protocol, a user can emulate MQTT clients (sensors, actuators, switches and so on) and design and execute performance testing.

Rely On Open Source and Collaboration

DevOps teams depend on robust communication and collaboration to avoid introducing technical debt. Sharing and communication are crucial if a team is attempting to accelerate delivery and improve quality through continuous testing. Teams need awareness of issues and status changes in order to work in concert—as opposed to creating rework and overhead for each other. This is especially true as each member of the team plays their specific role and works toward the overall goal.
Git Integration: Improved Team Collaboration for Agile Teams

One of the most important improvements to support the collaboration between Agile teams is the VuGen integration with Git. One of the leading source control management solutions. Git is a version control system that is used for software development and other version control tasks. As a distributed revision control system, it is aimed at speed, data integrity, and support for distributed, non-linear workflows.

VuGen integrates with Git Hub, allowing you to upload scripts from a Git repository and perform common actions such as Pull, Push and Track. You can also view history and manage your changes.

JMeter scripting tool support (Beta)

Starting with version 12.55, you can run your JMeter scripts in LoadRunner and Performance Center and integrate JMeter with additional script types in any performance test. This means DevOps teams can easily upload existing scripts created with a familiar tool (and obviously save time).

Use Tools That Reduce Complexity

Our testing and operational environments are becoming increasingly complicated. Tools can help move past this complexity and perform root cause analysis quicker. Anomaly detection is key to identifying anomalies and performance problems, as well as finding root causes using real-time metrics.
Quickly Isolate Problems
Anomaly detection enables you to quickly, intuitively, and effectively identify abnormal application behavior in performance tests. With version 12.55, you can now use anomaly detection built into Performance Center to accomplish those tasks. Engineers can use insights derived from the anomaly detection to speed their diagnosis and investigation into system performance—and even see the precise triggers that caused the anomalies.

![Anomaly detection](image)

**Figure 7.** Anomaly detection

Diagnose Design Problems
While browsing a customer app in a web browser, there are occasionally requests to servers that the user doesn’t necessarily like to include in the script, such as Google statistics, ads, third-party data, and so on. In LoadRunner and Performance Center version 12.55, VuGen introduces the new Recording Summary Report, which allows the user to:

- View general information on the recording session
- View detailed information on hosts, content types, and headers
- Conveniently manage traffic filtering (available in the Recording Options)

Information shown in the report allows the user to potentially make decisions regarding application design, by observing the number of requests, size of data sent and received, type of data, and so on.
Virtualization is critical to an organization’s ability to configure and test against many likely issues that customers will encounter during the course of an application’s lifecycle. The advantage of virtualization lies in constructing a complete environment that encompasses development, testing and operations, virtual services, virtual networks, and virtual data.

Figure 8. Recording Summary Report

This feature can help with troubleshooting issues (for example related to correlations found and replaced), leaving you the time to customize your scripts more efficiently. In addition, using the report makes removing these unwanted requests from the script much easier and faster.

Why You Need Software Lifecycle Virtualization

Virtualization is critical to an organization’s ability to configure and test against many likely issues that customers will encounter during the course of an application’s lifecycle. The advantage of virtualization lies in constructing a complete environment that encompasses development, testing and operations, virtual services, virtual networks, and virtual data.

If you can use network and service virtualization in your app development process, you will be more efficient and cost-effective, which means you will have a better chance at achieving key: beating the competition, reducing production incidents, and lowering costs.

We build in Service and Network Virtualization for continuous development and testing across teams even when services are not ready yet or constrained and where global network behavior creates obstacles to quality and performance.
Service Virtualization Integration

Often, an application under test depends on services such as API calls to other systems, whether external or internal. Because the development and testing teams don't control these outside services, the teams become dependent on availability of services to complete testing. There are many testing scenarios where a tester expects repeatable and reliable responses from a given service, but without control, this can be a challenge.

Simply said, external services can quickly become major bottlenecks for development and testing teams. To overcome this constraint, development teams can simulate or virtualize services so they can reduce dependencies and regain control. This allows them to test as much as they want—particularly load testing—without fear of affecting the performance of the real services.

To facilitate performance testing of business processes that contain services that are not available, LoadRunner and Performance Center integrate with Service Virtualization. This enables your team to test an application’s interaction with a variety of unique conditions, such as how the system behaves if a service is offline, or if it takes a long time to respond, or if its response is not what was expected (for example, an exception). Service Virtualization gives teams flexibility and complete control over test environments.

Network Virtualization Integration

Any application that relies on a network must be able to react and provide feedback to the user, even when the network is slow, erratic, or even unavailable. Network conditions, such as latency, bandwidth, packet loss, and jitter, must be taken into account when developing and testing software.

Network Virtualization (NV) is designed to enable precise network performance testing scenarios that deliver highly accurate and reliable performance results. Effective insights and analysis of those results are critical to bringing meaningful and cost-effective application or infrastructure changes.
To achieve superior application performance even in adverse network conditions, Network Virtualization is fully integrated with LoadRunner and Performance Center to simulate the application’s behavior under different network characteristics.

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. It 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, teams can view statistics that help you measure the quality of the user experience on your application.

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.

**Find the Right Application Testing Platform**

Application testing and quality assurance have never been so crucial. Your platform should keep pace with ever-increasing user expectations, accelerating release cycles, growing IT complexity, and massive scalability requirements.

Take a closer look at the new capabilities built into LoadRunner and Performance Center 12.55 to build, test, and deliver your applications with breakthrough speed and quality. Talk with your local representative to explore what your organization can achieve with version 12.55.

**Learn More At**

www.microfocus.com/loadrunner
www.microfocus.com/performancecenter