

What are the best Performance Testing Tools?

Edwin_Hernandez

August 1, 2018

In this post, we would like to talk about Performance Testing Tools. The market for Performance Testing has evolved over time with some newcomers and also test tools that have been around for 10-20 years and have been updating their features to rise up to the challenges of modern applications.

One important note is that I am only going to list full-featured Performance testing tools that are used for End-to-End Load testing, Endurance validation and capacity planning of Enterprise-level applications. With the popularity of Cloud computing, many of these platforms offer Load Tests integrated to their platforms (AWS, Azure, Google cloud). For the most part this is limited to URL or API testing, where you specify a list of URLs for the platform to use to stress a website and produce results. This only works on open websites that do not have authentication and do not need to be tested with complex Business processes (test cases). Some of these platforms have an option to upload a JMeter script or a VS webtest instead of specifying URLs. This will be covered below.

LoadRunner

LoadRunner is the oldest and most mature performance test tool in the market, it has been around since 1994, it really picked up on version 5 or 6 in 1998-9. This tool was originally created by Mercury Interactive, which was acquired by HP in 2006. Later in 2016 HP sold its software division to Micro-focus. LoadRunner is unique in that its core features are divided into 5 different software installations: VuGen, Controller, Generator, Agent process and Analysis. It has a wide range of technologies and communication protocols that it can test, however it is not very well adjusted to modern technologies like Cloud computing.

Pros:

- Some of its features are not matched by other competitors or not matched with the same level of quality, for example:
 - Its Analysis and Detailed Report features. LoadRunner is able to cross-reference, overlap metrics, do comparisons, with metrics from servers resources and load tests results. It can create fairly detailed Word Document Reports that are as close to be ready to be send away as it gets.
 - The ability to create complex test scenarios with groups of users starting at different times, ramping up independently. You can even make changes at run-time like increasing or decreasing the number of vusers on a specific script and know what each vuser is doing at a specific moment. There isn't another tool with these features out-of-the-box.
- It integrates well with SiteScope to monitor applications and HP Performance Center to schedule and pool resources to run multiple load tests in parallel.
- LoadRunner has the widest range of applications and protocols it can test. It is not only Web/HTML, also APIs, SOAP, Mobile, Databases, etc. LoadRunner protocols are libraries with custom functions that make scripting to test these technologies somehow easier and it requires less figuring out stuff (as it would be with JMeter or VS). Some of its protocols for ERP and CRMs encapsulate complexity and have ready-made correlations.
- Since it's a paid tool it comes with dedicated support, documentation, tutorials, etc.
- Supports Windows, Mac and Linux.
- It takes less time to setup and get started than other alternatives, notably JMeter.

Cons:

- It integrates with other HP ALM tools like UFC, SiteScope and HP Diagnostics, however HP/MicroFocus ALM suite somehow doesn't compare well to other more modern alternatives.
 - LoadRunner doesn't integrate well with third-party ALM tools. It integrates with Jenkins for Continuous-release scenarios, however it is not a seamless integration. Recently, LoadRunner is also able to run JMeter, Selenium and Nunit scripts but it requires an interface library and some manual configuration.
- Limited scripting language support. Until 2017 it was Ansi-C only. After 2017 LoadRunner can also be scripted with JavaScript. Ansi-C is not a OOP language.
- Just like other HP/Micro-focus tools, LoadRunner can be pricy. Not only the software itself is costly, but the protocols (web, citrix, SAP, etc.) are sold separately, need to be renewed frequently and the cost increases with the number of virtual users that your need to simulate (regardless of hardware, these are virtual user licenses).

JMeter

This tool by Apache has been around since 1999 and gained popularity throughout the 2000s, taking market share from LoadRunner mainly because it is a free, open-source alternative that can be expensively customized. It's important to point out that JMeter even though designed and used mainly for load testing, it is also used for Unit and Functional testing of FTP, LDAP, WebServices and database connections.

JMeter feels a bit dated because of its IDE and some of its features . Reporting is limited when compared to LoadRunner. Small companies love it not only because it's free and runs on any OS that supports Java but also because it can be used for unit test, functional test and monitoring which makes it kind of a swiss-knife of testing. This obviously attracts teams that want to save time and effort learning more than one tool.

Pros:

- Price. It's free, which also means that is not supported by a proprietor. However it is also open-sourced and has very good community and industry support whom have extended and fixed bugs on it throughout the years.
- Compared to LoadRunner, there are no licenses for protocols or virtual users, you can simulate as many users as your infrastructure can handle.
- It's large community support means that there is considerable training material, blogs, extensions, etc.
- Very flexible, it runs on any OS that supports Java: Windows, Linux, Mac, etc. Also records http traffic using any Browser (IE, Chrome, Firefox, etc.)
- Supports scripting with BeanShell, Groovy and Javascript,
- Has a wide range of protocols it supports: Web/http, WebServices, Databases, SOAP, etc.
- It has good integration with third party ALM tools like VSTS, HP tools, Jenkins, etc.
- It has a drag and drop declarative view for scripts, you can make changes to each request in a dialog interface. Which means that for simple test cases you don't need programming skills.. This somehow compared favorably with LoadRunner, but this feature is also offered by other tools (e.g. Visual Studio)

Cons:

- It is not the most user friendly tool, it is focused on the technology rather than making it easier to do things like defining an scenario based on business requirements or scaling the load.
- it uses different terminology than other tools (threads instead of v-users)
- It does the job (load testing) but has stated before it doesn't make things like scalability, coordination, integration, easy when compared to other tools.
- Reports are very limited, it generates XML, JSON and HTML logs that can be used for graphs but this feels more of a raw log than a report. This can be improved with extensions created by the community and industry, these extensions are not free though.
- As mentioned before it has good support of third-party ALM tools. JMeter however is not part of an Apache ALM tool stack which would make interactions between different components of the SDLC easier. It doesn't have an included out-of-the-box cloud load generation feature. Azure and AWS do take JMeter scripts for their own performance testing offerings.
- Some people advertise that scripting is not essential on JMeter, this may be true for simple Database or API testing but for long workflow based test cases on complex Web applications that require correlations, parametrizations to handle logins, cookies, sessions, etc., scripting is needed and JMeter doesn't make this easy compared to virtually every other alternative out there.
- Although it has good community support, it doesn't have proprietor's support for bugs, an official knowledge base or dedicated support.

Visual Studio Load Tests

Included by Microsoft in every release of Visual Studio since 2005. Its core features were updated in 2008 and 2012, since then most updates have been focused on Cloud Load Generation, Testing with VS Team Services and its integration with Azure and VSTS for DevOps and continuous release scenarios.

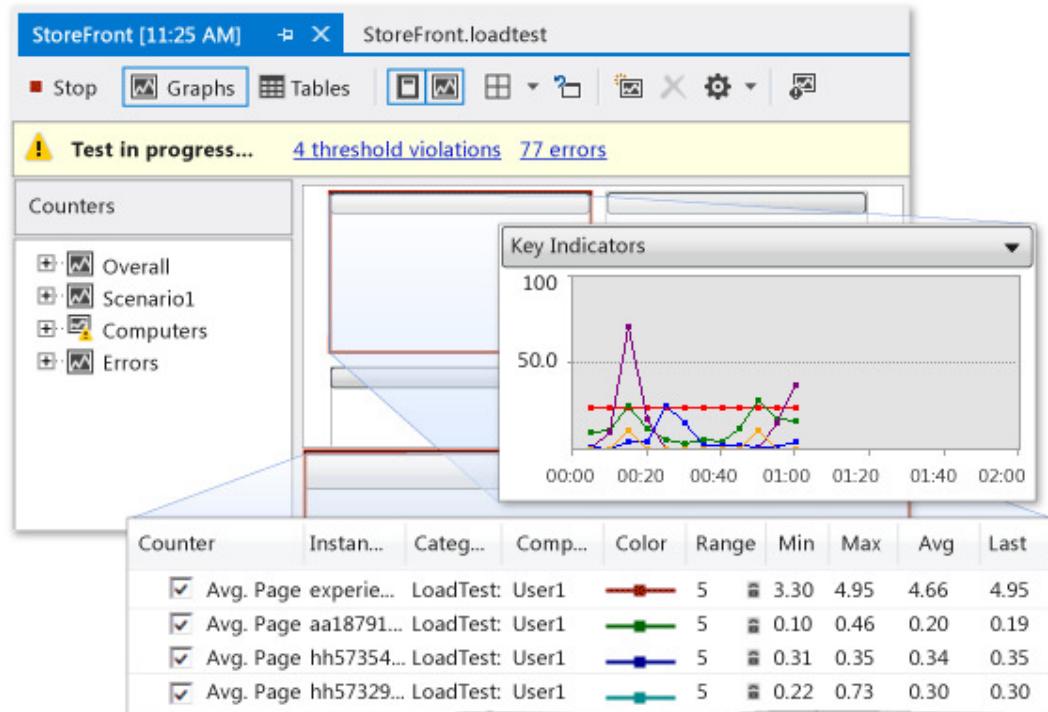
Pros:

- Load Testing is embedded into Visual Studio which makes it much more powerful than any other tool because it has access to VS features meant for developers such as line by line debugging, breakpoints, source control, collaboration, saving results to SQL, connections to databases, writing custom code, including external libraries, NuGet packages, creating extensions, customizing scenarios, management of cookies, sessions, etc.
- Virtual User Licenses. Similar to JMeter, VS Load Tests doesn't charge extra for virtual user licenses or protocols, you are free to simulate as many users as your on-premise hardware supports.
- On that point, scaling is supported and easy to use and setup. You install an Agent software on the load generation machines and a Controller software to coordinate it all. Alternatively you can use their integrated cloud load generation feature which is very easy to use. This requires an Azure subscription and it does have an extra cost based on cores and virtual users minutes (VUM), but you get a good amount of VUM with a free subscription.
- Very good integration with the Microsoft ALM tool stack: seamless integration with Azure and VS Team Services (VSTS) for continuous integration and continuous release scenarios. Reporting, results, tracking to requirements and builds can easily be managed in VSTS. Integrates also with VSTS for source control and Agile planning.
- You can load test directly from VSTS and Azure for simple test cases or upload a VS Load test.
- It has a good monitoring and analysis features that overlap and compare metrics from any Windows machine to which it has access. You get any metric that is accessible to PerfMon.
- It has a simple enough IDE that makes easy visualizing scripts (WebTests) in a declarative way, make edits to it, do correlations and extractions in the IDE without doing any scripting. Recording works really well and auto-correlations for supported applications like SharePoint make your life much easier.
- For more complex scenarios, programming skills are needed but the Visual Studio IDE makes it easier when compared to other alternatives. The IDE also makes easier creating Load Test Scenarios, managing connections to the Test Rig or Cloud Load Generation.
- Good support from Microsoft, there is a well-documented knowledge base (MSDN).
- Price. If you already have Visual Studio Enterprise or an MSDN subscription, that is all you need, there is not a separate license for the Load Testing feature, for virtual users, extra protocols, etc. If you use your own on-premise hardware for scaling the load, you can simulate as many v-users as you need.

Cons:

- The fact that it is part of Visual Studio makes it more powerful but also ties it a lot to developers. Most companies don't want to invest a Visual Studio Enterprise license on the testing team. Also the fact that it is not a standalone tool makes it a bit of an obscure feature that gets lost among the many features of Visual Studio. A good portion of the market doesn't know about its existence, hence the low % of popularity in the market share chart at the end of this article.
- As a feature of VS Load Tests, this tool supports recording and testing only with the Http/web protocol. However Visual Studio, being a development platform, can easily accommodate testing other protocols like Database, FTP, SOAP and even desktop and unit testing. This is because VS Load Tests can take unit test and codedUI class objects. For this however, you would need programming skills and know what you are doing.

- Limited support: It's Windows only, recording can only be done on IE and it supports scripting only with C#.
- Reporting is limited, VS Load Tests generate Excel reports for trends and comparisons but these reports are not as good as LoadRunner.
- Price. If you don't already have Visual Studio, the initial investment in a Visual Studio Enterprise License can be considerable. Once you have that, that is pretty much everything you need, as stated before Microsoft doesn't charge extra for virtual user licenses or additional protocols.



WebLOAD

WebLoad is a full-featured enterprise-level load testing tool by RadView. It has many similar features compared to LoadRunner or Visual Studio Load Tests while it has a lower cost than LoadRunner (but it's not free or open-source as JMeter is). It is a pretty popular LoadRunner competitor and has been gaining market in the last years.

Pros:

- It has a wide-range of protocols that it supports such as Http/web, SOAP, Databases, FTP, XML, SMTP, etc.
- It integrates with a good number of third-party tools for monitoring like AppDynamics, Dynatrace and supports mobile and UI test scripts from Selenium and Perfecto. Integrates with Jenkins for DevOps processes.
- Supports C and JavaScript for scripting.
- It has a full-featured IDE that is easy to use and had solid auto-correlation and edition features.
- Supports load testing on-premises and cloud load generation (unlike JMeter).
- Good customer support and documentation.
- Its analysis dashboard and Reporting features are very good, comparable to LoadRunner.

Cons:

- It has a better integration with third-part tools than LoadRunner but this integration is to App Monitoring and UI testing tools rather than to ALM or Continuous Release scenarios, so tracking to requirements, builds and bugs is not easy to achieve. Comparatively it is not part of a larger tool stack just like LoadRunner is part of the HP ALM tool stack and Visual Studio is part of Microsoft's ALM tool stack.
- Its range of supported protocols is better than Visual Studio and JMeter but lower than LoadRunner, it doesn't support Citrix, SAP, RDP, etc.
- Price. It costs about 50% of LoadRunner and has a similar protocol, licenses model which can be more expensive than Visual Studio.

Silk Performer

Originally created by Segue, acquired by Borland where it gained its most notoriety, it was acquired by Micro-Focus in 2009. (Micro-Focus now has two of the top offering in the Load Testing market, the other one being LoadRunner). It is a tool very similar to LoadRunner in features but these are not at the same level on quality or maturity.

Pros:

- Solid features comparable to LoadRunner and WebLOAD: consist of 3 components, Workbench, TrueLog and Performance Explorer, which are your scripting, test execution and analysis. It supports scaling the load thru load generators.
- Wide-range of protocols it supports: Web, Databases, Mobile, SOAP, FTP, Citrix, even ERPs. Comparable to LoadRunner.
- Analysis and Reporting is not as robust as in LoadRunner but much better than JMeter and Visual Studio.

Cons:

- The main drawback is that it uses a proprietary scripting language which means that it has an initial learning curve.
- Editing scripts, correlations and parametrizations are not as straightforward than LoadRunner or other tools in this article. It takes more time to get used to it and it's more complex to learn.
- Price-wise, it is a proprietor tool with a hefty price tag that is similar but perhaps a bit lower than LoadRunner. Its pricing model is similar though, you buy virtual user licenses, protocols and the analysis module.
- Monitoring system resources is not as easy when compared to other tools, it is restricted to a set of defined counters. LoadRunner provides more monitoring features.

Others

LoadUI by SmartBear

SmartBear has test suite that includes TestComplete for UI testing, SoapUI for API functional testing and LoadUI for load testing specialized on APIs.

SOASTA by SOASTA

SOASTA specializes in cloud load generation, a cheaper price model (when compared to LoadRunner) and faster scripting and testing. It is one of the most recent robust tools in the market.

Rational Performance Tester by IBM

Solid features, easy to use, embed in Eclipse, good for ERPs and IBM products. Limited Protocols, lacks on Execution and monitoring.

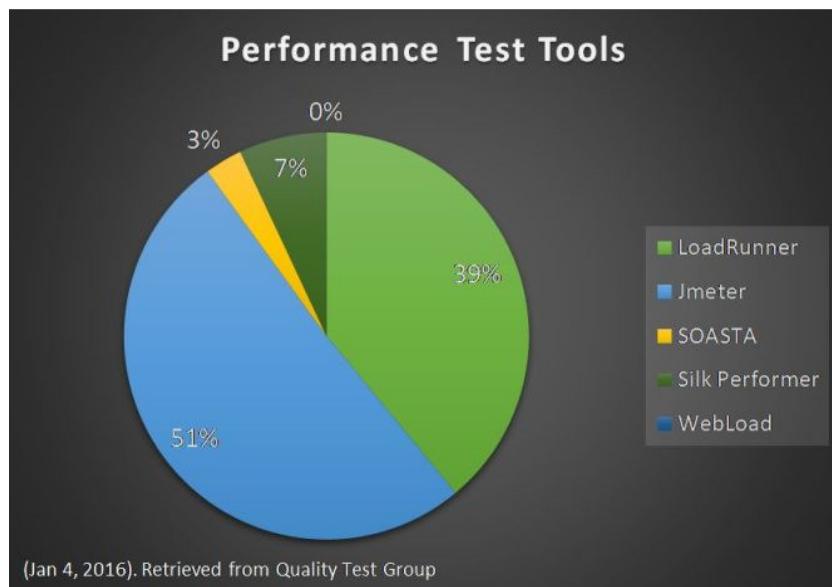
MARKET SHARE OF LOAD TESTING TOOLS

We struggled to find good market research on Performance Testing tools because of two factors:

1. There aren't many market researches and many of them are outdated back to when LoadRunner used to have 85% of the market.
2. Some researches mix Load Testing tools with UI Automation or Unit test tools (in the case of JMeter this may be valid, but you can't compare LoadRunner and Selenium).

Based on what I've found it looks like JMeter and LoadRunner are the two major players in Load Testing:

- JMeter specializes in small companies with very limited budgets that want keep using a particular programming language and want to integrate with other open-source ALM tools.
- LoadRunner is most popular with large companies that want to test enterprise-level applications like ERP and CRMs.



SO, WHICH ONE SHOULD YOU USE?

Consider the following:

- Do you need to align with a particular technology?
 - If your team is using HP or Microsoft ALM tools for Agile planning, continuous-releases or other DevOps scenarios, you may want to learn LoadRunner or Visual Studio. The same if you need to align with IBM.
 - If you are working in a cloud development. JMeter has the most third-party acceptance for Google Cloud, AWS and Azure. For Azure though, you may be better using Visual Studio since it provides the most benefits because of its integration to VS Team Services
 - Or if your team want to align with open-source technologies like Jenkins, JMeter is the best pick.
- What type of application do you need to test? API-based, WebServices, Http Web app?
 - Most tools in this list can accomplish the basic protocols but if you need to test an ERP/CRM and can afford the price, LoadRunner, WebLoad or Silk the most recommended.
- Consider your skills. Visual Studio is C# only, LoadRunner is C, JMeter and WebLOAD have the wider language range although they are Javascript-based.