

# How to Choose a Load Testing Solution

A Buyer's Guide

**RADVIEW**





# Table of Contents



Getting Started .....	3
Total Cost of Ownership .....	5
License Size & Cost.....	7
Buying Strategies.....	8
Product Demonstration.....	10
Feature Matrix .....	11
About RadView.....	13



# Getting Started

This paper will assist your evaluation process and selecting a load testing solution that is the right fit for your organization.

Many tools and options are available today for performing load testing. When coupled with the abundance of features and technologies that must be evaluated and compared, the selection process can be quite confusing. In addition to technological parameters, you must also consider issues like technical support, cost of ownership, and the skills of your testing team. This guide will help you navigate through these issues in a methodical way. We begin by explaining how to get started with the evaluation process. Then, we continue to discuss technical features to be evaluated, and finally, provide tips on how to negotiate your best deal.

The first step in choosing a load testing solution is evaluating your needs and resources. Here are some basic questions you should consider.



## How often do you run load testing?

Start by evaluating the frequency of running performance tests.

Will you be running tests once a quarter? Monthly? Daily?

If you run load tests often, then you'll want to look at a professional, higher-end type of solution. If you plan to load test infrequently, a "lightweight" or even open source solution may suffice.





# Getting Started



## On-Premise, cloud (SaaS) or hybrid?

---

Is your organization comfortable (from a security stand point) with a cloud solution, or do you need an on premise tool? Many solutions offer a hybrid approach where the load testing software runs on servers within your network, but also allow you to generate load via cloud based hardware (like Amazon AWS, Rackspace, etc.). Some lightweight cloud solutions are not capable of generating load on premise. Identifying your needs will eliminate many potential load testing solutions from your list.



## Which technologies and protocols?

---

Gather your list of requirements and then populate the spreadsheet matrix at the end of this document, to decide which features you need and the exact frameworks and protocols your development teams work with. This is an excellent way to evaluate and compare which vendor can best meet your needs. You can even add weight to the technologies or features so you can score solutions based on your needs.





# Total Cost of Ownership



Before we dive into license size and cost, let's touch upon TCO – Total Cost of Ownership. Total Cost of Ownership in load testing involves a few factors – from support to powerful features that should be built-in. You can get a free, open source tool. But you and your team won't have anyone to call, will be spending a lot of time reading and posting questions on sites like StackOverflow (great tool – but can you afford to rely solely on the public domain?).

There are lightweight tools which may offer good support, but will lack correlation and advanced reporting. If your load testing tool doesn't take you directly to the root cause and help you to identify bottlenecks, what good is it?





Do you need an open source, lightweight or powerful load testing tool? Consider these factors:

Category	Explanation
	<b>Support</b> You should expect easy access to tier 2 professionals during business hours. Support should help you and your team onboard and be very knowledgeable when it comes to scripting and scenario creation. When you can't rely on support, you will lose time and money on your side.
	<b>Scripting capabilities vs. GUI</b> A complete solution should offer a combination of an easy-to-use GUI and nonproprietary scripting language like JavaScript. The intuitive user interface will allow non-technical individuals to quickly create scripts. Without it, you will spend more money on training your non-technical staff. Scripting will allow developers and advanced load testers to address more complex scenarios. Look for tools that use standard scripting languages that also match the expertise of your team.





## Total Cost of Ownership

Category	Explanation
 <b>Automatic correlation</b>	An automatic correlation engine should identify and replace dynamic values which are unique for each run of the script such as session IDs, time, and others, and replace them automatically. Poor or no correlation is common in lightweight solutions. The money saved on a lightweight tool will be spent on labor-intensive script updates and manual correlation editing.
 <b>Integrations, frameworks and protocols supported</b>	A huge number of frameworks and protocols are available today to your development team. Consequently, you must ensure that your load testing solution supports the widest range of frameworks and protocols. Down the road, it's very likely that your development team will be using a different technology than today. A load testing tool that provides built-in integration with the technology used will make your testing much more efficient than tailoring an integration from scratch. Don't get locked into a long-term maintenance contract with a load testing solution that doesn't support a multitude of technologies.
 <b>Server side monitoring</b>	The purpose of load testing is to evaluate how your system behaves under intense usage. An enterprise-scale load test solution will offer a wide range of server performance monitors and technologies to collect data during load testing. Without server side monitoring you are missing a key piece of load test information.
 <b>Comprehensive reporting</b>	Reporting is often overlooked, but is often a weak area in many tools. However, extensive reporting capabilities will save you time and money identifying problems and performance bottlenecks. Ideally, reporting should provide a rich set of out-of-the box reports, but also the ability to easily customize reports so you can present data in any specific angle. Another important capability is a web interface to enable collaboration and the monitoring of results from any location in real-time.



# License Size & Cost

## Licensing Options

**There are two purchasing models that are common in load testing:**

### Perpetual License

You purchase and own the software with “x” number of virtual users. As long as you pay for maintenance, you receive support and upgrades. Even when you stop paying maintenance, you own the software and can keep using it up to the number of virtual users that you paid for. If you need more virtual users for one or more months, you can “rent” them.

### Subscription License

You do not own the software, but are “leasing it” for a period of time. Think of how Microsoft is now offering Office. You pay a yearly fee to use Office.

The subscription license usually includes all future upgrades and has no large down payment. The perpetual license costs more, but will save you money over the long-term. The devil is in the details – look closely at the small print and make sure you know all there is to know about any possible limitations and exceptions. Insist that the sales person explain all options in detail.



## Determining the number of virtual users

The license cost of most load testing tools will be determined by the number of concurrent virtual users you’ll be running, and to a lesser degree, the number of testers.

It is therefore important you carefully consider the number of virtual users to be purchased.

To determine the number of virtual users you’ll need, it is best to closely examine the number of real users you expect to use your system in peak times. For example, if you’re testing an e-commerce platform, you may look at the traffic during Christmas time (or any other peak time in your industry).

The number of virtual users for your load testing should be 30% above the expected number of users. This will ensure you can stress your system beyond expectations as well as address any future growth.

## Concurrent testers

Equally important is the question of how many team members will be using the load testing tool. For example, if agile development practices are used in your company and/or DevOps processes, it may be that you’ll want more users to be able to collaborate on load testing and be able to share results.



# Buying Strategies

Once you've selected your tool, your next step is price negotiation.

Here are 6 tips to help you save money and avoid unnecessary expenditures when procuring a load testing solution.



## **Larger buy upfront**

The larger the license of virtual user you buy, the more bargaining power you have when negotiating price. Also think about whether your management team will be willing to spend money again in one or two years from now. Management is often reluctant to spend money twice in a short period of time, so your best bet might be to consider purchasing future needs now. For example, the difference between a 1000 and 2000 virtual users might not be that large because the discount for 2000 virtual users will be higher.

## **Multi-year maintenance contract**

Another way to save money is to consider a multi-year maintenance contract. Usually, software companies will not discount on a single year maintenance contract. However, if you sign a multi-year commitment, you'll often be able to get discounts on maintenance cost.

## **Avoid hidden costs**

Make sure the price you get includes all features and options you need. Watch for hidden costs when procuring a load testing solution.





## Buying Strategies

### **Offer a case study**

Hi-tech companies are always looking for endorsements of their solutions. You can sometimes negotiate an additional discount if you agree to provide a case study after you've installed and used the load testing software for a period of time.

### **Maintenance fee percentage**

Maintenance fees are typically calculated as a percent of the list license fee and range between 15% and 20%. If your potential load testing solution provider is charging a higher rate, it may be negotiable, or you might want to look at an alternative solution.

### **Competitive upgrade**

If you find yourself unhappy with your current load testing solution either because it lacks features, the quality of support is unsatisfactory, or perhaps due to the high cost of the maintenance - you can sometimes leverage these issues in a new purchase. Some load testing solution providers will offer a competitive upgrade to certain competitor's products.

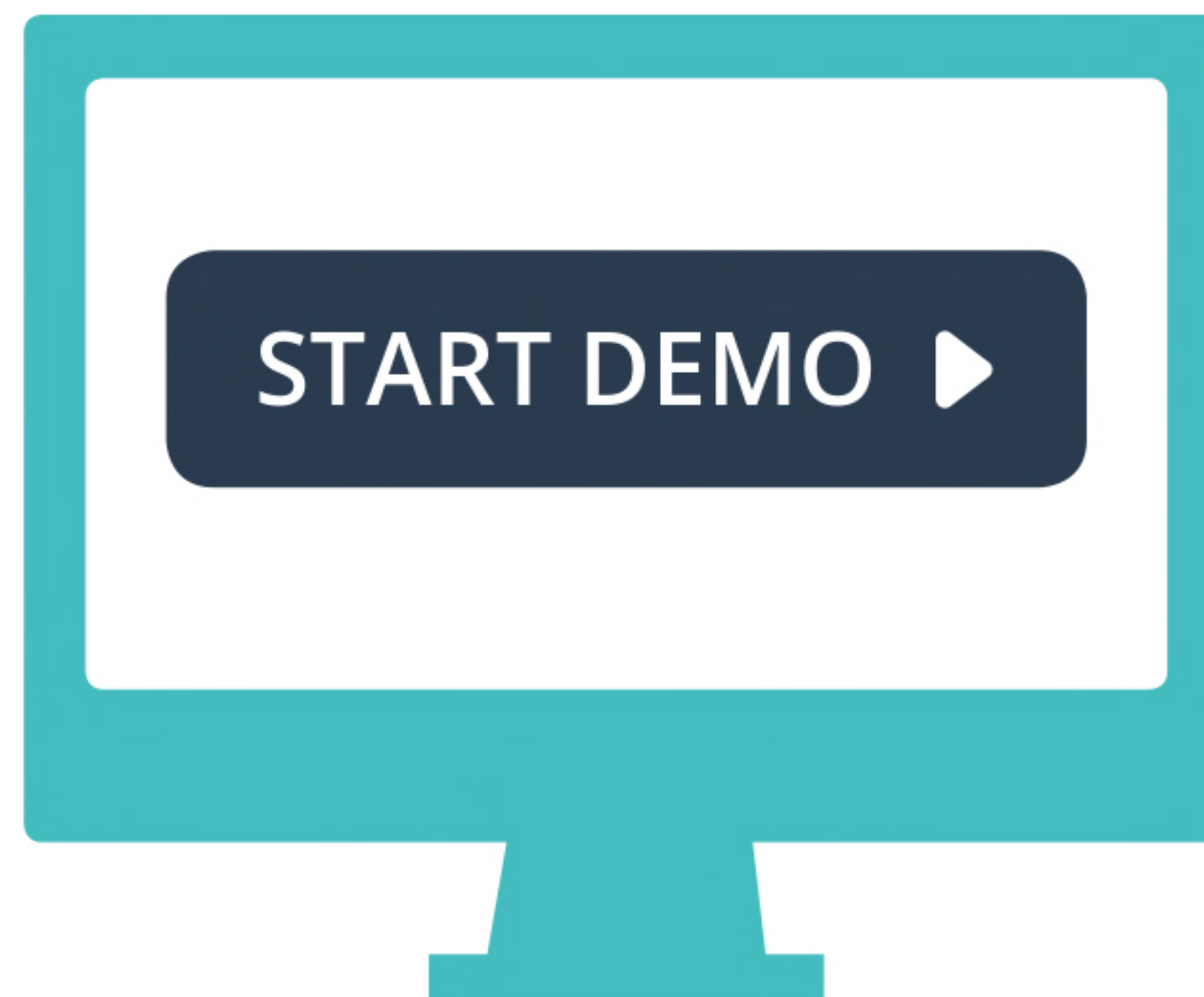


# Product Demonstration



Once you have boiled down your load testing options to two or three choices, getting an online demonstration by each of the vendors is a good idea.

Of course, you should also try the tools yourself, but a one-hour online demo can provide you a thorough functional overview of the capabilities and specific features. It will allow you to assess more accurately the capabilities of each of the tools, beyond just a checkmark indicating the existence of a feature.





# Feature Matrix

	WebLOAD
<b>Platforms</b>	
<b>Windows OS</b>	
User Interface (Controller)	✓
Execution Agent (Load Generator)	✓
Unix/Linux Execution Agent (Load Generator)	✓
<b>Protocols &amp; Technologies</b>	
<b>Web 2.0 &amp; Rich Internet Applications (RIA)</b>	
HTTP/HTTPs	✓
HTML5	✓
Adobe Flex/AMF	✓
XML/SOAP	✓
JSON	✓
Web Services	✓
WebSockets	✓
Rest API	✓
AJAX (Asynchronous JavaScript and XML)	✓
WebDAV (Web distributed authoring and versioning)	✓
Java over HTTP	✓
Push Technologies (streaming, polling, long polling, async requests, reverse ajax, comet)	✓
Kaazing	✓
LightStreamer	✓
Angular JS	✓
React	✓
Adobe Air	✓
SPA (Single Page Application)	✓
<b>Network Protocols</b>	
FTP (file transfer protocol)	✓
IMAP (Internet messaging)	✓
SMTP (Simple mail transfer protocol)	✓
POP3 (Post Office Protocol)	✓
JMS	✓
LDAP (Lightweight Directory Access Protocol)	✓
TCP/IP, UDP	✓
DNS (domain name service)	✓
Telnet	✓
NNTP	✓
SSH (Secure shell)	✓
<b>Authentication</b>	
SSL	✓
TLS	✓
NTLM	✓

	WebLOAD
Kerberos	✓
Basic	✓
<b>Databases</b>	
JDBC (Java database connect)	✓
ODBC (Open Database Connect)	✓
Oracle	✓
Microsoft SQL Server	✓
PostgreSQL	✓
MySQL	✓
MDB	✓
<b>Applications</b>	
SAP NetWeaver	✓
SAP Web, Web Dynpro	✓
SAP BusinessObjects	✓
Infor Lawson	✓
Microsoft Dynamics	✓
Ellucian Banner/Luminis	✓
PeopleSoft	✓
PrimaVera (Oracle)	✓
JDEdwards	✓
Oracle Applications – Oracle Forms	✓
Oracle 2-tier	✓
Oracle e-business suite	✓
Oracle Siebel	✓
Sage	✓
Kronos	✓
Salesforce	✓
Oracle Hyperion	✓
Sharepoint	✓
HR Access	✓
IBM Maximo	✓
EMC Documentum	✓
<b>Server Technologies</b>	
J2EE	✓
.Net	✓
Node.js	✓
PHP	✓
Apache	✓
Tomcat	✓
IIS	✓
NGINX	✓

	WebLOAD
Weblogic	✓
Websphere	✓
ColdFusion	✓
JBoss	✓
JOnAS	✓
Oracle GlassFish	✓
<b>Content Management Systems</b>	
WordPress	✓
Drupal	✓
Joomla	✓
Magento	✓
<b>Wireless &amp; Mobile</b>	
iOS	✓
Android	✓
Windows Phone	✓
Blackberry	✓
<b>Multimedia (Voice &amp; Streaming)</b>	
RTMP	✓
RTSP/RTP	✓
<b>Performance Monitoring (Server-Side) - System Resource</b>	
Microsoft perfmon	✓
Linux/Unix SSH	✓
Linux/Unix rstatd	✓
Solaris	✓
IBM-AIX	✓
HP-UX	✓
<b>Performance Monitoring (Server-Side) - Databases</b>	
Microsoft SQL Server	✓
Oracle	✓
ODBC/JDBC	✓
PostgreSQL	✓
MySQL	✓
<b>Performance Monitoring (Server-Side) - Protocols</b>	
PerfMon	✓
SNMP	✓
SSH	✓
JDBC	✓
RSTATD	✓
JMX	✓



# Feature Matrix

## WebLOAD

Performance Monitoring (Server-Side) - Application/Web Servers	
ANY via JMX	✓
ANY via SNMP	✓
Adobe LiveCycle ES	✓
Ariba (SAP procurement)	✓
ATG Dynamo (Oracle eCommerce)	✓
Broadvision (eCommerce)	✓
ColdFusion (Adobe)	✓
Fujitsu INTERSTAGE	✓
GlassFish AS (Oracle)	✓
IBM WebSphere (5, 6, 7, 8.5)	✓
IPlanet (NAS)	✓
JBoss (RedHat)	✓
JOnAS	✓
Microsoft Active Server Pages	✓
Microsoft ASP.NET	✓
SAP NetWeaver	✓
SilverStream (Oracle Application Server)	✓
WebLogic	✓
Apache	✓
Microsoft IIS	✓
SunOne (Sun + NetScape)	✓
Microsoft Windows Media Server	✓
Real Networks RealServer	✓
Integration & Extensibility	
Command Line Interface	✓
Open Architecture (Public SDK/API)	✓
3rd Party Tools Integration/Plug-in	
AppDynamics	✓
Dynatrace	✓
New Relic	✓
Nagios	✓
PerfectoMobile	✓
Jenkins	✓
Selenium – browser Emulation	✓
Ranorex – browser Emulation	✓
Original Software TestDrive – browser Emulation	✓
Git	✓

## WebLOAD

Interoperability/Extensibility	
Java	✓
COM	✓
General Features	
Protocol-Level Testing	✓
Browser-Level (Emulation) via Integrations	✓
Distributed Architecture	✓
Page Elements/DOM Access	✓
JSON Object Access/Parser	✓
XML Object Access/Parser	✓
Java Object Access	✓
Real Page/Browser View	✓
Full HTTP Headers View	✓
HTML (Response) View	✓
Response Validation	✓
Cookies Support	✓
Cache Support	✓
Multiple IP Address (IP Spoofing)	✓
Network Emulation	✓
Recording Features	
Browser-based recording	✓
Native Mobile Recording	✓
Insert Transactions during recording	✓
Developing/Debugging Environment	
Access to the script	✓
Quick scripting building blocks	✓
Intellisense/Auto-completion editor	✓
Context-sensitive Help/Tooltips	✓
Debugging	
Breakpoints	✓
Watch (Variables Value)	✓
Step-by-Step	✓
Scripting Language	
JavaScript	✓
Java	✓
Automatic Data Correlation of dynamic values	✓
Manual Data Correlation of dynamic values	✓
Parameterization	
From files, Strings, Numbers, Dates	✓
Random, Sequential, Unique	✓

## WebLOAD

Synchronization/Rendezvous Points	✓
Think Time (Sleep)	✓
Transactions	✓
User-defined Timers	✓
Test Configuration & Scheduling	
Goal-Oriented Load Testing	✓
Load Scheduling Profiles	
Constant	✓
Linear	✓
Random	✓
Intervals	✓
Steps	✓
Ramp Up	✓
User-defined	✓
Test Execution	
Real-time server-side monitoring	✓
Real Time View of ALL Performance Measurements	✓
HTTP statuses reporting and logging	✓
Integrated Reporting Manager	✓
MIX execution	✓
Cloud load generation	✓
Probing Client	✓
Throttle control	✓
Freezing tests during execution	✓
Send Notification on defined triggers	✓
Test Results Analysis & Reporting	
Web Dashboard	✓
Reporting Engine and Analytics	✓
Dynamic and easily customizable reports	✓
Predefined Reporting Templates	✓
Custom/user-defined Reports	✓
Reports Exporting and Publishing	✓
Comparison and Regression Analysis	✓
Support & Maintenance	
Online (customer portal)	✓
Expert level (tier 2,3) support on a first call	✓
Phone	✓
Email	✓
Documentation, tutorials, help files, Best Practices	✓
Trainings	✓



## About RadView

---

RadView is a pioneer of web and mobile load testing solutions. Since 1993 the company has been offering WebLOAD, an enterprise-grade software for large-scale load testing, helping companies launch internet applications with confidence. Deployed at leading organizations, WebLOAD offers a best-value load and performance testing solution.

**RADVIEW**