5 Urgent Signs

It's time to re-evaluate your performance testing toool

RADVIEW

Introduction

Way back, your load testing tool selection made perfect sense. It answered your technical requirements and the cost fit your testing budget.

But time passed and things have changed. New technologies have emerged and your load testing tool is not the perfect match anymore. Sometimes it's simply an issue of inflated costs; in other instances, technical support has become frustrating; or possibly, you discover that you're spending precious time on manual tasks, which should have been automated.

Here are five signs that may point out that it's time to consider an alternative to your existing load testing solution.

Software Maintenance Fees are Too High

Years ago, you may have gotten a great deal for your software license. But now when maintenance is up for renewal, you discover it's eating too much of your budget and you're not feeling you're getting your money's worth.

High software maintenance costs may be related to any of the following:

Overpriced software license

Overpriced software license
The load testing tool you have may be top of the line - but are you paying for too many features, protocols and technologies you don't really need?

Increasing number of virtual user licenses

As web technologies have matured, many companies are discovering they need to test their system under a much larger user load than in the past. Typically, this greatly affects your license costs.

Large number of licenses

Your team has grown over the years and now you want to provide access to developers, DevOps personnel and QA team members.

A-la-cart features

Your system under test has incorporated new technologies you need to test. Yet, you are being charged for each additional technology added, which significantly increases your total costs.

You're Spending Too Much Time on Manual Adjustments

2

There are many load testing tool choices - both low end and high end - and each might be perfect for a different set of needs. But if you discover you're spending too much time on manual tasks and adjustments, you might be using the wrong tool.

One example is test scenario maintenance

You might discover that you're spending too much time manually updating scenarios or re-recording, simply because your tool does not extract assets such as .css, or .js files every time they are changed. Another sign is that you're spending too much time detecting scripting errors due to manual changes.

Another example is correlation & parameterization.

If you need to spend hours or even days manually adjusting correlation parameters, you may want to look for a more efficient solution.

Reporting is another task that should not consume any of your time.

If you're not getting the reports you need and find yourself exporting test results to Excel (or another tool) to manipulate test metrics, then something is wrong.

Identifying Root-Cause is Taking Too Long

At the end of the day, performance testing is about identifying bottlenecks and improving your system's performance. If your load testing tool lets you stress your system but does not help you much with analyzing results, quickly identifying performance bottlenecks and pinpointing the location of the problem, then perhaps you're not fully achieving the goal of performance testing.

You might want to explore the option for a tighter integration between the load testing tool and an APM (application performance monitoring) tool that will allow you to accelerate the process and implement a faster turnaround time of testing and software deployment.

3

Technical Support is Frustrating

By now you're aware that performance testing is a complex process that always presents new challenges.

- When you do need support, is your vendor helping you deal with issues
- How long do you need to wait for technical answers
- Are you getting answers quickly, or do you need to pass through first-level support teams that are not proficient enough to answer your questions

You Cannot Accommodate All Stakeholders

4

Way back, software testing was the property of the QA group only, with testers running tests and reporting results back to R&D. **The emergence of new processes like agile development, continuous integration and DevOps has changed the way testing, development and software deployment are performed.**

This means, for example, you should be able to share test results with a larger number of people from multiple groups. You may also want to store test data in a shared repository, rather than a single tester's desktop.

Perhaps you also want to view results in real time, while tests are running so that you can detect problems, pause the test and make quick fixes instead of discovering issues much later.

And last but not least, you may want to automate your performance testing and incorporate it into a continuous delivery process.
If you find that your load testing is still confined to a lab and cannot be shared by multiple groups and stakeholders, it's time to re-examine your solution.

Identifying Overloaded Load Generators

Regardless of whether you decide to switch your performance testing tool or not, this is a bonus section, which you may find useful while running load tests.

A common problem in performance tests is overloaded load generator machines, which can skew test results. Below are tips that will help detect whether your load generator machines are overloaded:

Load generator machine resources

- CPU utilization and memory usage
- Context switches per second - a high context switching number indicates that the CPU is less efficient, spending more time on itself than performing its task
- Page faults - when this number is high, the system is spending resources on writing data to disk, indicating inefficiencies and potentially harming performance
- Disk queue length - while a queue for writing to the disk will always exist, if this number constantly increases it can indicate load generator overload. If queue length increases while load size remains constant, the indication is even stronger.

Transactions per second

- Compare the load generator machine with a 'probing client' machine - a separate machine that executes a single virtual user. Assume you increase the load size and see that TX/sec value is not growing linearly. By examining the TX/sec created by the probing client and comparing it to the load generator machine, you can determine whether your load generator is having trouble.

Try WebLOAD

Are you overpaying for unused features? Are you working too hard to get the results you expect from performance testing?

RadView WebLoad is the sweet spot of performance testing.

RADVIEW