

A White Paper Analysis from Orasi Software

# Strategies for Performance Testing Custom Salesforce Deployments



# Contents



---

## 3 Executive Summary

---

## 4 Meeting Expectations

### Performance Testing Complications

Lightning Experience  
Complexities

Multi-Tenant Environment

---

## 5 Performance Testing Complications

Scheduling Conflicts

Salesforce Governors

Stress Testing Prohibitions

---



---

## 5 The Lightning Experience

---

## 6 The Lightning Experience

Testing Considerations

---

## 7 Optimizing Salesforce Performance Testing

Benchmark the Environment

---

## 8 Optimizing Salesforce Performance Testing

Adopt Testing Best Practices

---



---

## 9 Optimizing Salesforce Performance Testing

Acquire Appropriate  
Resources

---

## 10 Optimizing Salesforce Performance Testing

Obtain Test Plan Approval

Monitor and Remediate

---

## 11 Toward a Better Solution

## 12 About Orasi Software, Inc.

---

## Executive Summary

In many organizations, web and mobile applications now compose the majority of development efforts. Performance can make or break the success of these deliverables, and users are increasingly unforgiving. For these reasons, performance testing is critical, yet the unpredictable nature of app usage, device variability, and other factors makes it more complicated, as well.

For organizations using and customizing third-party (3P) software as a service (SaaS)—and Salesforce in particular—performance testing becomes even trickier. With any SaaS, deploying organizations have little to no control over the 3P app servers and their loads, increasing the difficulty of running tests and evaluating results.

Salesforce ups the ante, requiring organizations to test in a sandbox wholly under its control—and restricts testing to times it designates to alleviate impacts on other users. Performance test plans require advance evaluation and approval (for more about the process, see Obtain Plan Approval, in this paper). This is a logical approach, given the potential impact of unrestricted testing on other users, but it complicates the work of testing teams ensuring processes and queries run quickly and without incident, even with custom code injected.

Fortunately, teams can achieve a reasonable measure of confidence in performance through perseverance, ingenuity, and specialized test approaches and systems. To do so, they should be working in a fully mature test environment, and their testing resources—whether in-house or outsourced—should be seasoned professionals experienced testing under these unique conditions. This white paper will delve into strategies that can help companies attain these goals and ensure success with their custom Salesforce deployments.

---

*With Salesforce, deploying organizations have little to no control over app servers and their loads, must test in a sandbox wholly under Force.com control, and are restricted to testing at times that alleviate impact on other users. Highly seasoned Salesforce testing experts, operating in a mature test environment with access to robust testing infrastructure, can minimize these obstacles and ensure the best possible performance.*

---

## Meeting Expectations

Perceptive organizational leaders recognize that performance isn't optional, given that user expectations continue to accelerate, with negative perceptions of slow systems wreaking havoc:

- Revenue for Microsoft's Bing search engine dropped nearly five percent when tests delayed search-result delivery by two seconds.
- A two percent downtime metric for a business application or web-based service equates to 45 hours of lost productivity over a year.
- Per a 2015 survey by Dimensional Research, 61 percent of mobile users expect apps to start within four seconds, 80 percent will abandon a problematic app if any issues aren't resolved after three tries, and 53 percent will uninstall an app that crashes, freezes, or has errors.

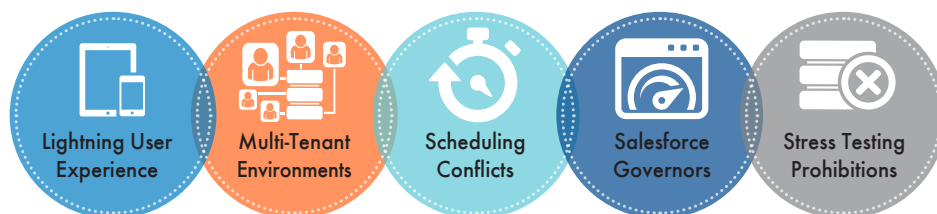
Admittedly, enterprise users may not have the option of abandoning Salesforce or a related mobile app if it doesn't function as expected. They will be less likely to use it, however, thereby reducing productivity. If the browser experience is also problematic, operating efficiency will diminish even further. With a CRM platform like Salesforce, a drop in worker efficiency and system usage can cause a reduction in customer contact, sales opportunities, and corporate revenue.

## Performance Testing Complications

Salesforce and its experts have published several documents reassuring software teams of the reliability and high performance of Force.com. (Read one, which is still substantially relevant, [here](#).) The performance of Force.com is not at issue, but speed and reliability of the core platform does not guarantee the performance of customized deployments. Additionally, limitations enforced by salesforce.com Customer Support pose significant performance testing challenges.

---

### SALESFORCE—PERFORMANCE TESTING CONCERNS




---

### Lightning Experience (Lightning) Complexities

Salesforce has recently released its new user experience, Lightning. The consumer-like experience is modern, efficient, intuitive, and broadly compatible, but it also increases testing complexity compared to the Salesforce Classic interface (more about that in [The Lightning Experience](#)).

### Multi-Tenant Environments

As mentioned earlier, user activities and transactions for all licensees are executed on Salesforce's app servers. It is therefore impossible for organizations to know how many commands are being executed, how many users are working, or the total load capacity of the servers compared with expected load at any given time. (Salesforce does not provide specifications for the system or log data on total system activity or loads.)

## Scheduling Conflicts

Organizations must include a proposed test schedule in performance testing plans submitted to Salesforce for approval. Salesforce.com Customer Support, and not the Salesforce customer, decides whether these times are acceptable. Obtaining approval for a schedule that is workable—for both Salesforce and the customer—often involves negotiation and multiple requests.

## Salesforce Governors

To ensure inefficient code for processes and queries doesn't monopolize the shared resources of a multi-tenant environment, the runtime engine in Apex, Salesforce's programming language, strictly enforces limits. If code exceeds a limit, the associated governor issues a runtime exception that cannot be handled, shutting down the process. The governors track and enforce a number of limits, including per-transaction limits, size-specific limits, Force.com platform limits, and more.

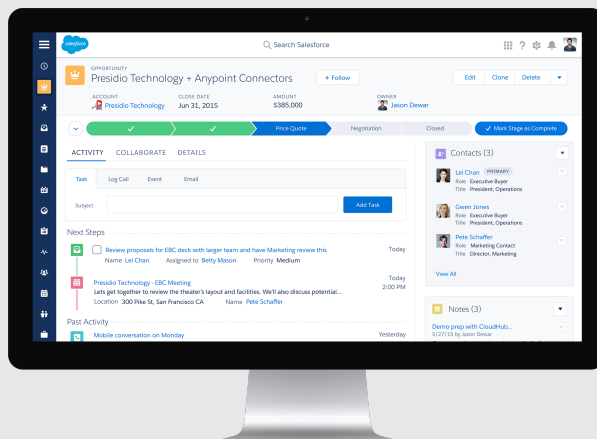
Salesforce governors complicate performance testing—but they also illustrate why it is so important. Properly managed, performance testing helps organizations determine if their business logic will trigger governor limits, incurring charges from Salesforce and/or a poor user experience.

## Stress Testing Prohibitions

Stress testing is fundamental to ensuring performance in the most demanding conditions, but Salesforce doesn't allow it. Even if it were allowed, its value would be diminished given that customer organizations have no insight into how Salesforce handles load balancing or manages the activities of an unknown number of customers in each multi-tenant environment.

## The Lightning Experience

The Lightning Experience is designed to give Salesforce users an advanced experience across every device—desktop, tablet, and mobile. Pages have been redesigned, processes have been streamlined, and workflows are more intuitive, enabling users to see more relevant information, more quickly, with every view or query.



Despite its appeal, Lightning is not for everyone. Per Salesforce, it is optimal for organizations with a sales-focused operating model and technologically sophisticated sales teams. Additionally, as of this writing it had not achieved the core functionality of Salesforce Classic. (View the latest update on the limitations of Lightning here.)

*(Continued, next page)*

## The Lightning Experience (continued)

Unless an organization is already planning a Salesforce upgrade and is prepared for extensive change management, the most pragmatic plan can be to defer adoption until all desired Salesforce Classic features are present. When enterprises do proceed with the upgrade, they should be prepared for extensive testing.

Although Salesforce reports most existing customizations should remain intact through Lightning adoption, there is no guarantee. Before upgrading, companies should confirm existing business processes and test cases are in working order and that existing customizations have been fully tested and all deficiencies remediated.

### Testing Considerations

- Salesforce offers a mechanism for users to switch between Lightning and Salesforce Classic, but this does not minimize the need for testing the new user experience.
- Since customizations may break during the Lightning upgrade, firms should retest them as part of the deployment.
- Although Salesforce Classic can be tested headless, Lightning cannot. It requires GUI testing with a tool such as Micro Focus TruClient, which will in turn require a load-generating system. Organizations will need access to powerful load generators, either in-house or cloud-based, to handle the testing loads.
- The broad compatibility of Lightning will necessitate extensive browser and device performance testing, including the latest versions of Chrome, Firefox, Safari, and Edge. (This directive is especially applicable to organizations building custom apps.) Companies who move to Lightning must ensure their performance testing tools utilize the latest versions of these browsers.

“

The Lightning user experience requires GUI testing with specialized tools, and effective performance testing will involve powerful load generators.

”

Salesforce offers several tools, primarily the Lightning Design System, Lightning App Builder and Lightning Components, to help companies modify or customize Lightning and to extend or build responsive, performance-optimized web apps for mobile and desktop devices. These tools will not eliminate or minimize the requirement for performance testing, and they may encourage more custom development, which will in turn necessitate additional testing.

In short, organizations planning to migrate to Lightning—with or without customizations—should prepare for extensive testing, with performance testing being paramount.

## Optimizing Salesforce Performance Testing

Salesforce management recognizes that enterprises need to ensure performance for their Salesforce customizations. However, it also asserts that most organizations need to performance test only if they have highly customized code or large transactional volumes. It suggests that companies should be able to achieve suitable performance through Force.com performance profiling tools. (The Force.com performance profiling toolset is designed to help teams create, debug, and test applications. Read more about them, as well as the Salesforce performance profiling methodology, [here](#)).



Although Salesforce stipulations prevent any company from being 100% certain of high performance, comprehensive performance testing can ensure performance at a reasonable level.



---

This toolset, especially the Developer Console and Workbench when used in concert with debug logs, will help teams identify performance issues associated with Visualforce pages and Apex classes. (Visualforce is the component-based user interface framework for the Force.com platform.) However, the Salesforce recommendations tend to oversimplify the effort required to achieve a satisfactory outcome.

To ensure reliability and high performance, teams must ensure that custom application or business logic meets desired response times and that estimated transaction throughput of anticipated loads is accurate. Restrictions on accessing application servers or testing under real-world conditions can be largely overcome if activities are effectively planned, coordinated, and executed.

Although Salesforce stipulations prevent any company from being 100% certain of high performance, comprehensive performance testing can ensure performance at a reasonable level. In the next few sections, we will detail some of the key elements for success with such an effort.

### **1** *Benchmark the Environment*

Before running load tests, run benchmark tests to establish a baseline for the results of all future load scenarios (e.g. load, peak, endurance, etc.). Baselines also provide an opportunity to identify problems early in the testing cycle, as high transaction response times at baseline are a predictor of even bigger problems under heavier loads.

Optimally, tests should be run with a single user per script—and no other testing or users should be running on the test environment concurrently—to capture system response times at an idle (resting) state. With Salesforce, benchmark tests must run in the shared environment, making it impossible to capture resting system response times.

To minimize the impact of load variations, run the test multiple times and at different points during the day and night. Assuming the results are reasonably consistent across the board, results from the median baseline report will be appropriate for the benchmark.

## 2 *Adopt Testing Best Practices*

Achieving an acceptable level of performance with customized Salesforce deployments requires a test strategy and test plan that support the most robust testing possible given the limitations of Salesforce. Adopting testing best practices is fundamental to achieving these goals.

### **Overall Program Management**

- Ensure business requirements and use cases are clearly and appropriately defined.
- Adhere to sound test data management practices—with functional data requirements, data requests, and a repeatable process.
- Incorporate a fully implemented automation framework and develop/manage scripts effectively. Adopt service virtualization if it is not already present.
- Allow time for extensive browser testing, and ensure tools used are up to date. Salesforce recommends users access Force.com through (and, officially, only supports) the latest versions of Chrome, Firefox, and Edge. It no longer guarantees or supports functionality in Internet Explorer.
- Initiate and follow a quality improvement program using test metrics to build actionable KPIs with reasonable targets for reducing defects over time. (Read about the methodology, [here](#).)
- Plan for ongoing performance testing, including updating of scripts and potentially testing tools, which should occur not only with subsequent releases of in-house customizations but also with each major Salesforce update.

### **In the Testing Lab**

- Root out inefficient business logic during unit testing.
- Use Salesforce profiling tools to identifying performance roadblocks before performance testing. Any slow-performing queries or custom code identified in the profiler will presumably perform even worse under heavier loads.
- Incorporate inbound and outbound WSDL calls to 3P applications (integrations) into scripts, then validate them through low-volume tests, comparing resulting metrics with the baseline. Integrations are a common source of performance issues and must be thoroughly performance tested.
- Use a load generator to simulate concurrent users and put the application through real-life loads, varying test loads to simulate as many conditions as possible.
- Schedule tests effectively within Salesforce parameters.
- Obtain fresh baselines before introducing new or updated code, whether driven by an internal change or a Salesforce update. Baselines provide the data to ensure that new or updated code does not introduce bugs or increase transaction response times.



### 3 Acquire Appropriate Resources

Load testing for Salesforce customizations—especially browser and app testing—can place extreme loads on servers. The stated core requirements for load generators, which are indispensable for these efforts, are often misleading. For example, the core requirements for Micro Focus TruClient indicate it can run on a dual-core, 2.2 GHZ processor with 8 GB of RAM and 50 GB of drive space.



For Salesforce load tests, Micro Focus TruClient will require significant resources to obtain meaningful results, and companies' servers must also be capable of handling the test loads.



---

For Salesforce load tests, Micro Focus TruClient will require far greater resources to generate the users needed to obtain meaningful results, and companies' servers must also be capable of handling the loads placed on them during the tests. Enterprises could need dozens of powerful servers yet only require such capacity intermittently, reducing the ROI of acquisition and maintenance. In such situations, enterprises often license cloud-based solutions or work with an organization that maintains a high-capacity environment—and the manpower to perform tests.

Personnel also must have the skillset to develop, implement, and run an effective, ongoing Salesforce performance testing program that addresses not only testing for initial deployments but also for Salesforce release updates, which occur at least three times per year. It is often far less expensive to source these high-value professionals from a consulting organization than to hire them or attempt to train existing team members. For its customers, Orasi provides consultants—software specialists and engineers—with expertise testing Salesforce and navigating its processes and requirements.

Companies that use outsourced personnel can achieve even greater value by contracting with a software services firm that packages performance testing experts with full-service test planning, execution, management, and quality assurance. For its customers, Orasi provides both technology (e.g. load generators and the load testing environment) and human (Salesforce specialists that can go on site) resources.

4 Obtain Test Plan Approval

Performance test plans must be submitted to salesforce.com Customer Support, along with production organization IDs and contact information, at least two weeks prior to proposed test dates. Per Salesforce documentation, the test plan should include:

- A description of the tests
- Record counts of the objects in the tests
- The test scripts and tools (e.g., Micro Focus TruClient; Micro Focus LoadRunner) to be used for tests
- Estimated testing loads, framed in transactions per second
- Testing dates and times

ELEMENTS OF A SALESFORCE TEST PLAN\*

1	2	3	4	5
Description of the tests	Record counts of the objects in the tests	The test scripts and tools to be used for tests	Estimated testing loads, framed in transactions per second	Testing dates and times

*\*Two weeks in advance of testing (minimum), organizations must submit completed items, along with production organization IDs and contact information.*

Salesforce.com Customer Support may accept plans, request modifications, or reject them. A rejection or modification request can delay planned activities, wasting resources and causing release dates to slip. For the best outcomes, organizations should ensure plans will meet Salesforce requirements before submitting them or have skilled third-party negotiators to intercede on their behalf.

5 Monitor and Remediate

Best practice is to monitor the resources of every system that is part of the overall architecture and business processes affected by the application under test (AUT). This enables the testing team to determine how the system reacts during a performance test and whether or not the test is successful. However, Force.com system resource information is confidential and may not be available to your team. At a minimum, monitor all in-house and 3P servers integrated and communicating with Force.com.

Furthermore, once salesforce.com Customer Support has approved the test plan and teams have put it into action, they will monitor the tests. If any test puts too much load on the system, from Salesforce’s perspective, it will be shut down.

Consequently, as loads increase with each round of testing, your team should compare the results with projected metrics. (Salesforce.com Customer Support can provide metrics from the salesforce.com logs to help with post-test analysis.) If results indicate out-of-range loads, halt testing until the cause can be identified and remediated. Such an approach can prevent salesforce.com Customer Support from aborting tests due to excessive loads on Force.com.

## Toward a Better Solution

With the Force.com platform, as with any software program, achieving the best possible performance given prevailing conditions requires more than performance testing. It requires adherence to a quality-assurance-focused performance optimization program.

Developing such a solution for Salesforce, in-house, is not a fast or easy task. Organizations must navigate the intricacies of obtaining test plan approval, maintain robust load testing resources, manage a best-practices testing environment, and uphold an action-oriented, monitored, analytics-based feedback loop that promotes continual improvement.

Enterprises that expend the resources and effort to succeed gain the additional benefit of a meaningful quality improvement initiative. Organizations seeking a less burdensome—and potentially less expensive—alternative reap the same value with hosted or cloud-based testing platforms staffed by Salesforce testing specialists. Companies such as Orasi offer turnkey, expertly staffed testing labs that can handle virtually every aspect of testing customized Salesforce deployments, including Salesforce negotiations and problem resolution.

To delay or avoid choosing one path or the other dramatically increases the risk of ongoing post-release performance failures, sanctions by Salesforce, and extreme user dissatisfaction.

---

“

With the Force.com platform, achieving the best possible performance requires adherence to a quality-assurance-focused performance optimization program.

”

---

## About Orasi Software, Inc.

Orasi is an award-winning software reseller and provider of software training, support and professional services. To help companies focus on a complete software quality lifecycle, Orasi offers market-leading consulting services and solutions to support automated testing, application performance management/intelligence, mobile technologies, DevOps pipeline efficiency, and operational excellence. Orasi continues to expand its offerings across the entire software delivery spectrum, from data analytics to continuous delivery and open source tooling. Orasi maintains strategic partnerships with Micro Focus (formerly HPE), Chef, Delphix, SAP, XebiaLabs, and others. For more information, please visit [www.orasi.com](http://www.orasi.com).



For more information, contact Orasi today.  
[www.orasi.com](http://www.orasi.com) | 678.819.5300

© 2018 Orasi Software, Inc. The information contained herein is subject to change without notice.

Orasi is a trademark of Orasi Software, Inc. All other product and company names are trademarks or registered trademarks of their respective owners.