

DATASHEET
.....

Application Load Testing Service

Proactive Readiness for Critical Shopping Events

The sudden growth in ecommerce has made it difficult for application owners to predict traffic volumes to their online storefronts, and this is likely to continue in 2021 as well. To be ready for the holidays, or any special marketing event, merchants need to understand their application's readiness to handle increased surges of traffic. Research shows that online consumers go elsewhere if they deem a site too slow (taking more than 3 seconds to load) or if they experience availability issues, such as a failing checkout. While the Webscale platform provides multiple caching layers, and proactively auto-scales application servers, there are numerous architecture and code dependencies that impact a site's ability to scale under significant load.

It is therefore of critical importance to understand:

- How the code behaves under heavy load.
- What functionality or web pages are slow.
- At what point, if any, do errors start appearing for the users.
- How the user's web experiences change over increasing load.

Conducting a thorough evaluation of your site's performance at peak demand ensures that any critical issues, particularly around scalability, are avoided. It is especially critical to do all of the above before the high traffic shopping events, which is why Webscale introduced its Application Load Testing Service.



Webscale Application Load Testing Service

Webscale's Load Testing Service is run as a synthetic test across any staging or cloned production site, hosted in any cloud provider or on-premise data center. The purpose of the service is to simulate end user patterns and shopping behavior on the application, at a higher scale, while measuring its behavior, responsiveness and overall performance.

How it Works

The Webscale Application Load Testing Service executes a series of tests against a staging site or, in the case of a Webscale managed application, a Webscale hosted clone of your live production website. The service uses traffic viewer access logs or collects requests from real web server logs, understands user behavior and traffic patterns, and replays these against the site, as if they came from the original requesting IPs to the pre-production deployment. However, in a test scenario, these requests are replayed at an accelerated scale, so the application is processing a significantly increased volume of traffic and mimicking real user shopping patterns. The service then captures these results using the Webscale Customer Portal (shown as the “Loadtest-app” application), with the traffic logs available to view in the Webscale Traffic Viewer, for as long as the temporary loadtest-app is available. A report is also made available, on-demand, to describe the tests, the patterns, and the results.

Testing is typically divided into two phases, and depends on the subscriptions of the load testing service:

- **Phase 1:** The system as originally deployed is tested for capacity using anonymous users (non-logged in users). This is a replay of accelerated traffic without add-to-cart and checkouts. This establishes a baseline capacity from the system, can identify code and infrastructure deficiencies during load, and captures the bulk of load generated by high user traffic.
- **Phase 2:** The testing tool adds the ability to have logged-in users, add-to-cart and a high volume of checkouts to the baseline traffic provided in Phase 1.



Phase 1: Baseline Load Testing

A number of key measurements are recorded during Phase 1, with a goal of identifying average response time to the requests submitted, and determining what issues may be causing any anomalies in the results.

Some of the tests carried out in Phase 1 include:

- Increasing the load test volume by increasing traffic samples taken from the logs, and executing multiple runs of the Webscale Load Testing tool on the access logs provided.
- Amplifying traffic by a minimum of 10x during the test overall using a sliding scale.
- Monitoring CPU load on database.
- Setting response time goal and monitoring total requests made, & average response time during test.
- Monitoring for any scaling issues, and ensure that application response time remains within test parameters.



Phase 2: Cart Load Testing

To ensure your storefront can not only handle a surge in traffic, but also continue to process large volumes of simultaneous checkouts, Webscale can also carry out load testing on your shopping cart.

Some of the tests carried out in Phase 2 include:

- Generating a checkout load test by recording an actual browser session where a product is added to cart, checked out, the payment POST request is successful, and checkout success page* is reached with a 200 response code.
- Setting checkout rate according to, and beyond, peak checkout rates identified from past Black Friday and Cyber Monday sales events.
- Ensuring testing is limited by traffic generation only, not infrastructure capacity.
- Performing tests as if all traffic has been served from origin. Once the site is live, there will be less traffic coming to the origin due to CDN offloading.

* In order to facilitate checkout page load test, the application will need to support non-payment gateways or “fake” non-fulfillment product SKUs