MAINTAINING FAST AND CONSISTENT PAGE LOAD TIMES AT PEAK DEMAND ON THE WEBSCALE PLATFORM

INTRODUCTION/EXECUTIVE SUMMARY

A customer website was subjected to 3.5 hours of load testing that scaled from 1,000 users per hour up to 33,000 users per hour. During this load testing:

- Over 250,000 HTML requests were processed by the site, culminating in 3976 checkouts.
- The average page load time remained below three seconds.
- Once peak load was reached, the application cluster had scaled out to 11 instances, servicing 145,000 page views per hour with an hourly checkout rate of 2640 checkouts/hour.
- From Google Analytics data, peak production traffic for the customer was 14,000 page views per hour with a checkout rate of 276 checkouts/hour.

TESTING PARAMETERS

The balance of the load testing traffic was based on customer provided analytics regarding bounce rate and total checkout percentage, as follows:

- View only users (view home page, category page, three products): 55%
- Cart abandonment users (same as view only, plus add the three products to cart): 37%
- Guest checkout users (same as cart abandon, plus guest checkout with promo): 8%

Each category and product page is randomly chosen for each visit to the site. Load testing began at 1,000 users per hour and increased by 1,000 users per hour every six minutes until the peak arrival rate of 33,000 users per hour was reached 198 minutes later.

ARCHITECTURE

The architecture consists of one data server and a scaling application cluster.

Data Server – m3.xlarge (4 vCPU, 15 GB RAM)

- MySQL 14.14 Distrib 5.6.30
- redis 2.8.4-2
- NFS 1.2.8-6ubuntu1.2

Application Cluster – c3.xlarge (4 vCPU, 7.5 GB RAM)

- Apache 2.4.7-1ubuntu4.9
- PHP 5.5.9-1ubuntu4.16

In front of the application cluster were three Webscale application delivery controllers (ADCs). They provide load balancing between the scaling application servers, optimization of image, JS, and CSS resources, CDN integration, and a tier of resource caching.

TESTING METHODOLOGY

To drive the traffic, three JMeter nodes were created. Each node ran a modified version of the Magento Performance Toolkit, customized for the customer's website. Each JMeter node was directed at an individual ADC. To better represent a customer experience, the JMeter nodes were created in the Google Cloud while the test application deployment was deployed in AWS.

TESTING RESULTS

To establish a baseline for comparison, we ran the same JMeter test suite used in load testing against the current production site at a 100 users/hour rate during off-hours, running all steps up to (but not completing) a checkout. This baseline of 2.44 seconds can be used as a comparison for all following average page load time results, and is included for reference when appropriate on the following figures. Note that JMeter does not execute any JavaScript, and so this page load time will be lower than experienced in a web browser.

The impact of increasing load on the system and the effect on average page load time was the most important metric measured in this load testing. As seen in Figure 1, the average page load time remained under three seconds for the entirety of the test, and under the baseline production reference of 2.44 seconds for 99.2% of the testing period.

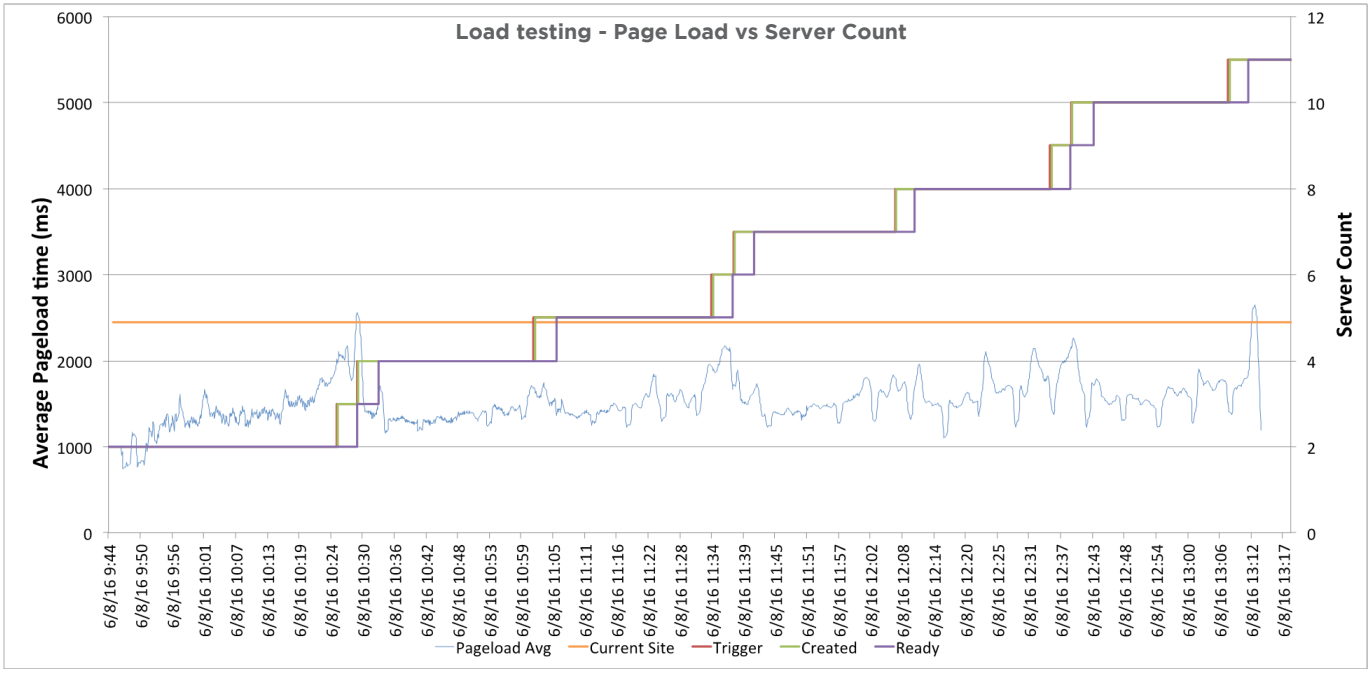

Figure 1: Average Page load time versus Server Count

At each scale event, we observed a consistent scale out time of 3.5 minutes per scale request. The time required to scale out measures how long AWS takes to create a functioning application server that is ready to service production traffic. In Figure 1, we can see that the average page load time remained relatively constant throughout the test. The regular fluctuation observed in the graphed results occurs on a regular six minute interval and is an artifact of the test.

The relationship between page views per hour and server count was observed to be a linear relationship as seen in Figure 2. This indicates each server that was added to the application cluster was able to handle the same amount of traffic as each server already in the application cluster.

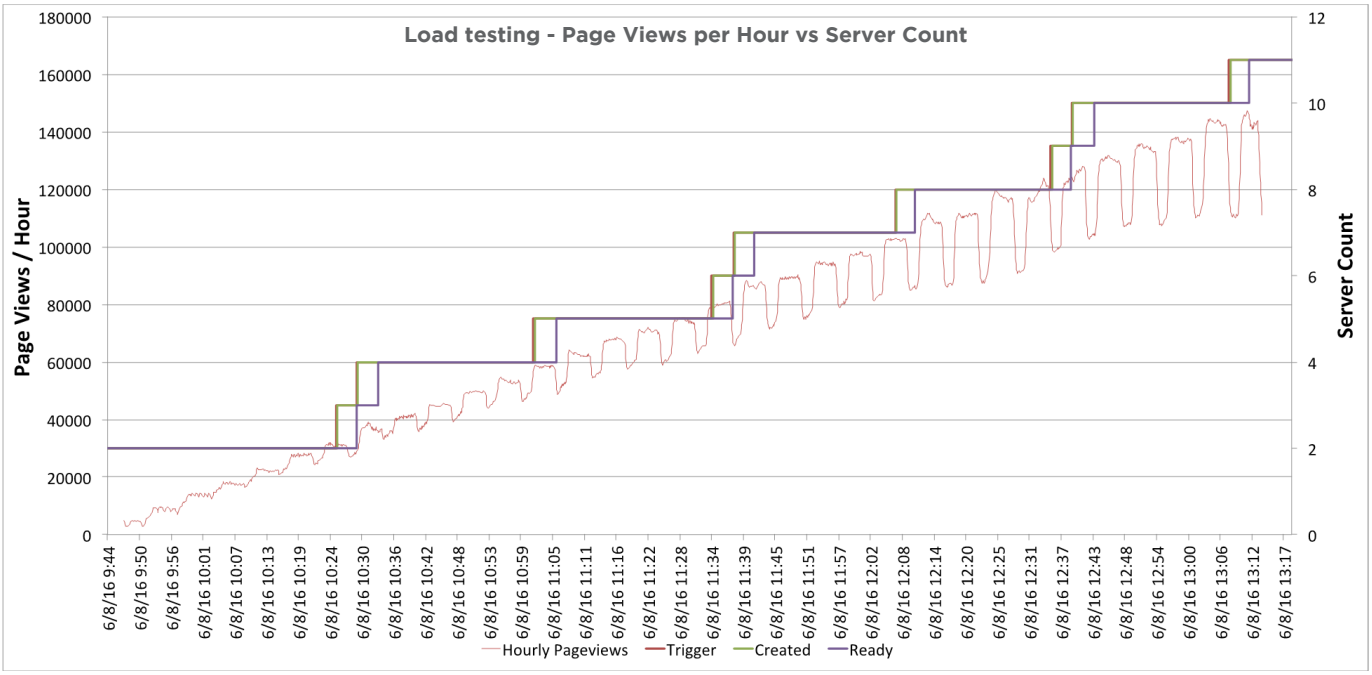

Figure 2: Page Views per Hour versus Server Count

As the load testing increased, the hourly checkout rate increased at the architected rate of 8% of total traffic. As seen in Figure 3, the average page load time remained consistent while this rate increased to a final value of 2640 checkouts per hour.

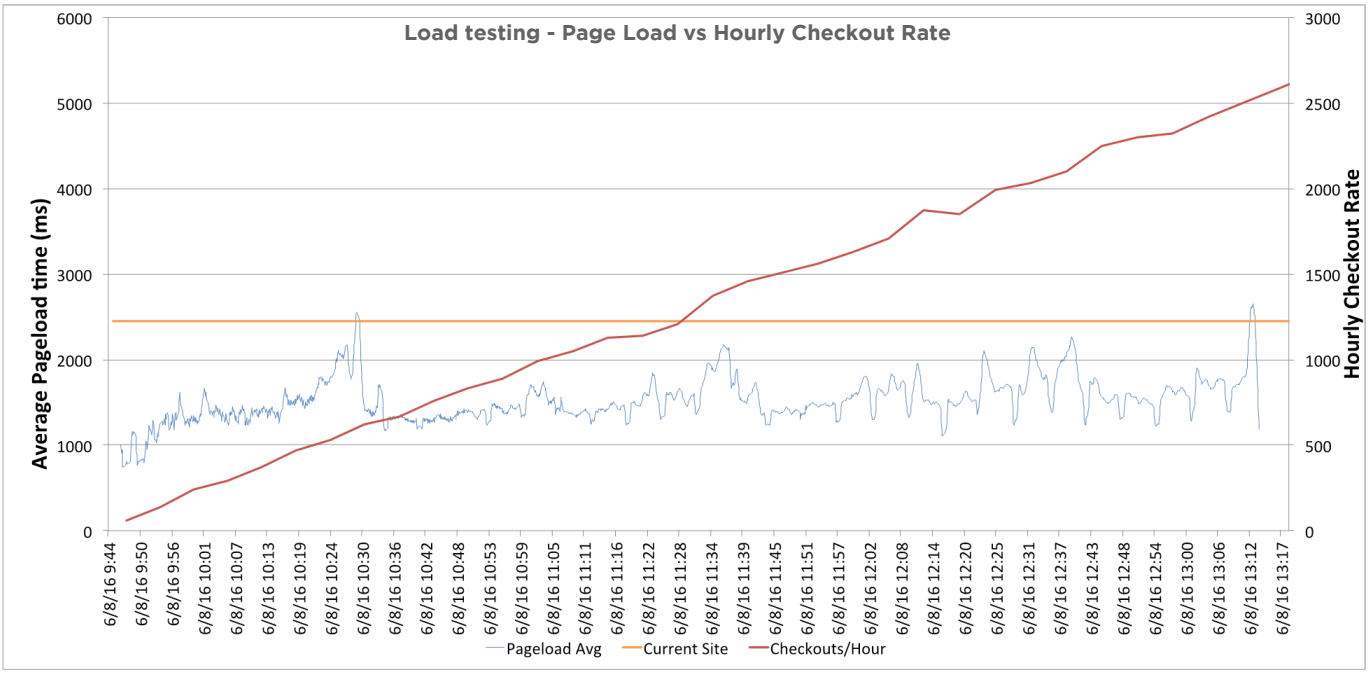

Figure 3: Average Page load time versus Hourly Checkout Rate

CONCLUSION

Based on the results of the load testing described in this document, the scalable application architecture is shown to deliver a consistent user experience for the customer's Magento e-commerce platform at traffic and checkout levels exceeding 10 times the measured peaks.