Load Test Report

Date: 5/28/2015

Test from : virginia

Query URL: http://reviewsignal3.flywheelsites.com/

Started at: Thu May 28 2015, 12:42:48 -04:00

Finished at: Thu May 28 2015, 12:43:48 -04:00

Test link: https://www.blitz.io/to#/play/input/virginia:50174c781f6ce4ff8e7a2f7af8e92f12

Analysis

This rush generated **49,960** successful hits in **60 seconds** and we transferred **648.26 MB** of data in and out of your app. The average hit rate of **833/second** translates to about **71,942,400** hits/day.

The average response time was **30 ms**.

You've got bigger problems, though: **10.11%** of the users during this **rush** experienced timeouts or errors!

Hits **89.89%** (49960)

Errors **6.47%** (3596)

Timeouts **3.64%** (2022)

Response Times	Test Configuration	Other Stats
Fastest: **24** ms	Region: **virginia**	Avg. Hits: **833** /sec
Slowest: **140** ms	Duration: **60** seconds	Transfered: **7.54**MB
Average: **30** ms	Load: **1-2000** users	Received: **640.72**MB

Hits

This rush generated **49,960** successful hits. The number of hits includes all the responses listed below. For example, if you only want **HTTP 200 OK** responses to count as Hits, then you can specify **--status 200** in your rush.

Code	Type	Description	Amount
200	HTTP	OK	49960

HTTP 200 OK **100%** (49960)

Errors

The first error happened at **45 seconds** into the test when the number of concurrent users was at **1498**. Errors are usually caused by resource exhaustion issues, like running out of file descriptors or the connection pool size being too small (for SQL databases).

| Code	Type	Description	Amount
17	TCP	Connection reset	17
23	TCP	Connection timeout	3579

Connection reset **0%** (17)

Connection time... **100%** (3579)

Timeouts

The first timeout happened at **57.5 seconds** into the test when the number of concurrent users was at **1916**. Looks like you've been rushing with a timeout of **1000 ms**. Timeouts tend to increase with concurrency if you have lock contention of sorts. You might want to think about in-memory caching using redis, memcached or varnish to return stale data for a period of time and asynchronously refresh this data.
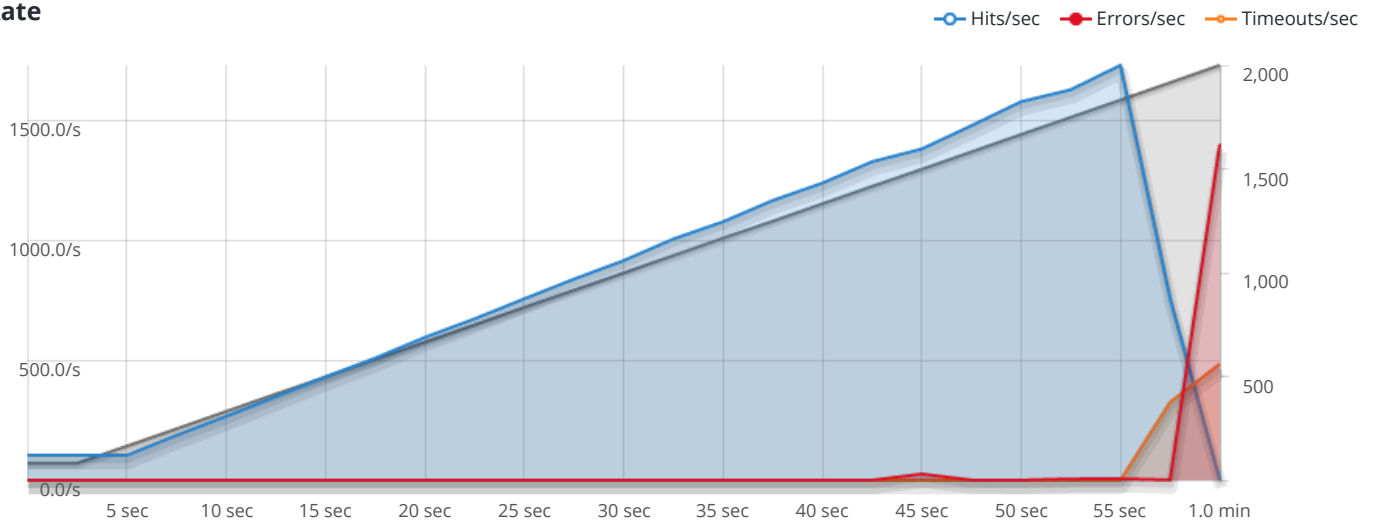
Response Times
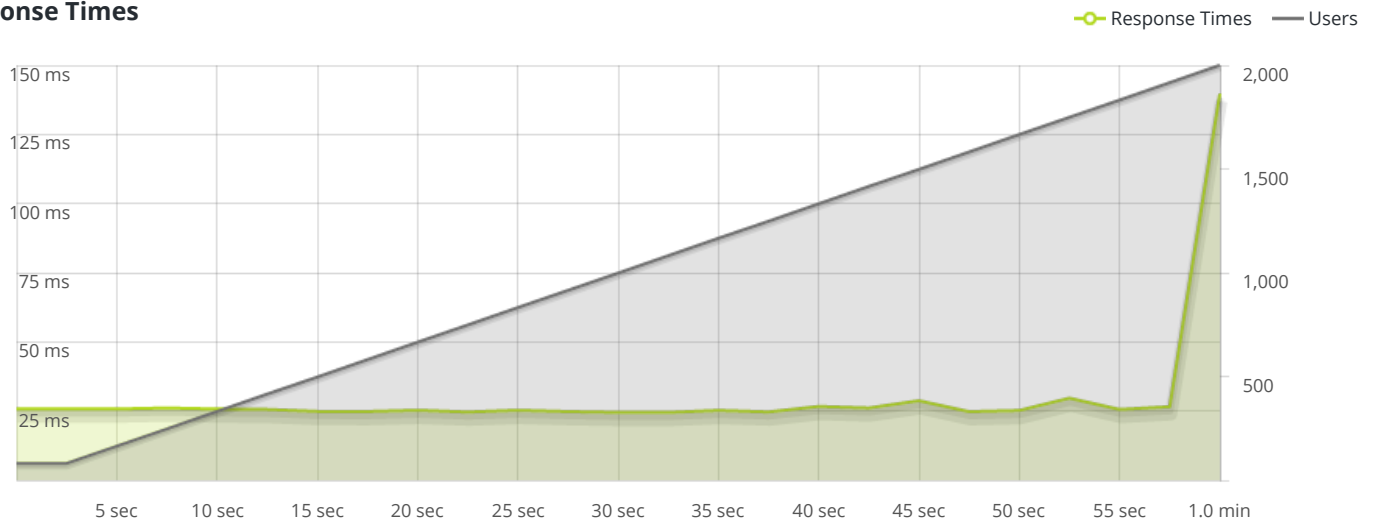
STEP 1
―O―Response Times

The max response time was: **139 ms @ 2000 users**

Hit Rate

STEP 1
―O― Hits/sec   ―O― Errors/sec   ―O― Timeouts/sec

The max hit rate was: **1,728 hits per second**