Load Test Report

Date: 7/18/2016

Test from : virginia

Query URL: http://testdomainkevin4.com/

Started at: Mon Jul 18 2016, 04:36:55 -04:00

Finished at: Mon Jul 18 2016, 04:37:55 -04:00

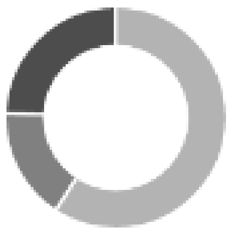
Test link: https://www.blitz.io/to#/play/input/virginia:b3d4ce86fef76f73f20089f491a41191

Analysis

This rush generated **28,568** successful hits in **60 seconds** and we transferred **478.42 MB** of data in and out of your app. The average hit rate of **476/second** translates to about **41,137,920** hits/day.

The average response time was **929 ms**.

You've got bigger problems, though: **40.81%** of the users during this **rush** experienced timeouts or errors!



Hits **59.19%** (28568)

Errors **24.35%** (11753)

Timeouts **16.46%** (7945)

Response Times	Test Configuration	Other Stats
Fastest: **192** ms	Region: **virginia**	Avg. Hits: **476** /sec
Slowest: **1,889** ms	Duration: **60** seconds	Transfered: **5.19**MB
Average: **929** ms	Load: **1-3000** users	Received: **473.23**MB

Hits

This rush generated **28,568** 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.

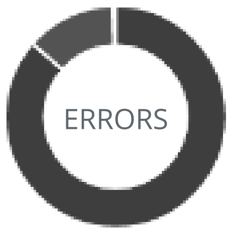
Code	Type	Description	Amount
200	HTTP	OK	28568

HTTP 200 OK **100%** (28568)

Errors

The first error happened at **20 seconds** into the test when the number of concurrent users was at **997**. 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	10161
23	TCP	Connection timeout	1544
		Response duration overlimit	48



Connection reset **86%** (10161)

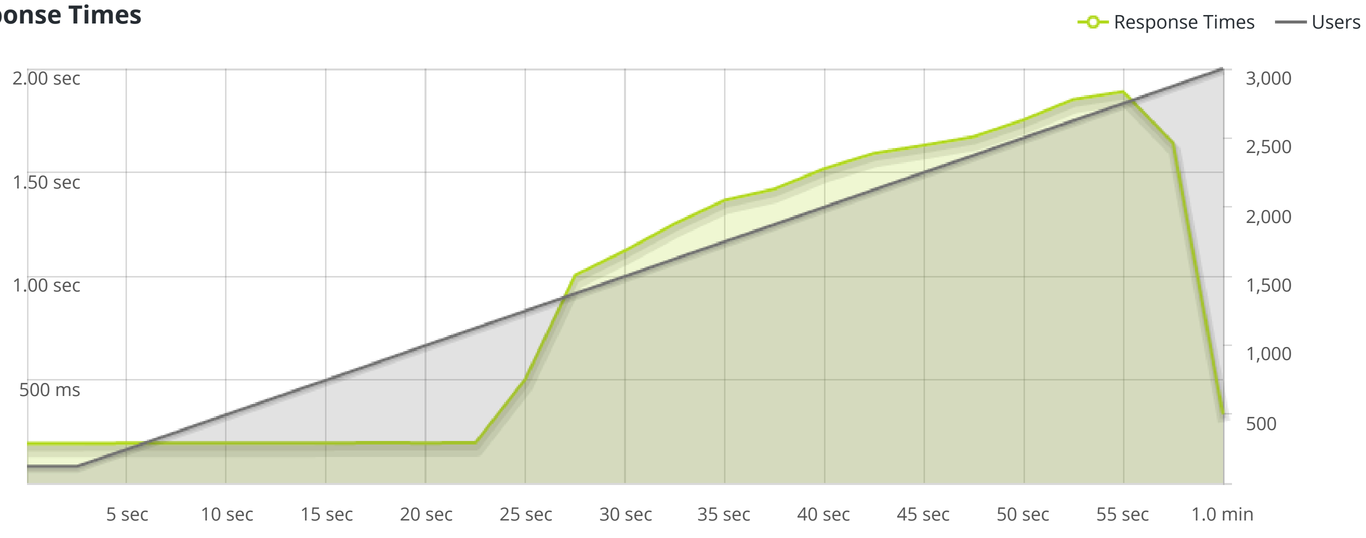
Connection timeo... **13%** (1544)

Response duratio... **0%** (48)

Timeouts

The first timeout happened at **25 seconds** into the test when the number of concurrent users was at **1247**. 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.

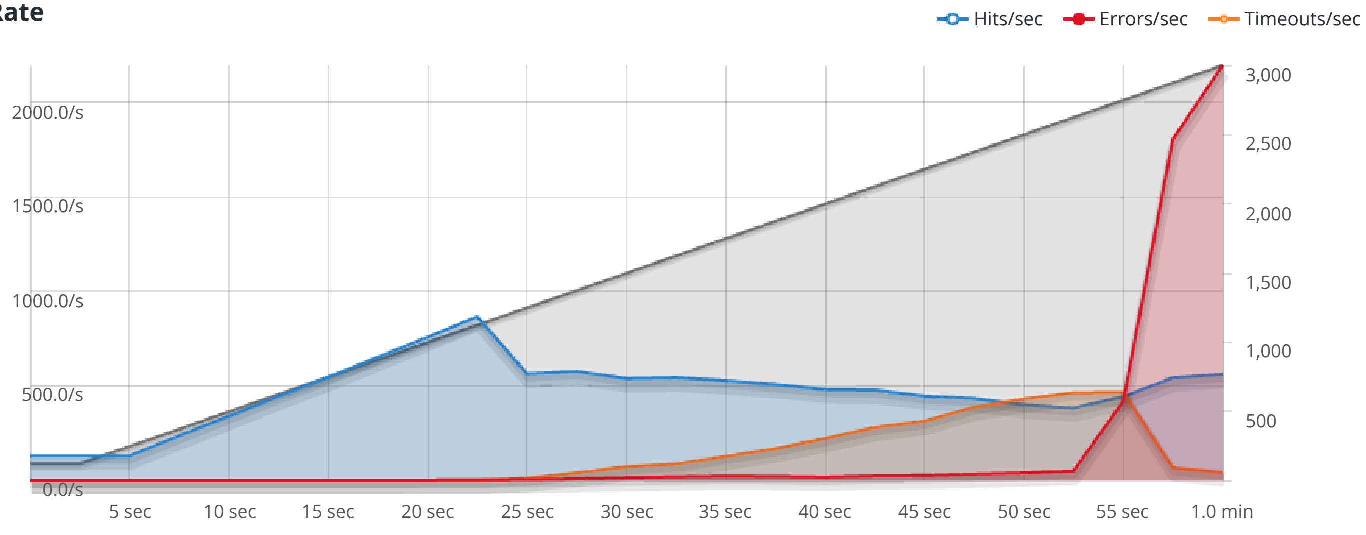
Response Times



STEP 1
Response Times

The max response time was: **1889 ms @ 2748 users**

Hit Rate



STEP 1
Hits/sec Errors/sec Timeouts/sec

The max hit rate was: **866 hits per second**