LOAD TEST REPORT

DATE: 6/26/2014

TEST FROM : VIRGINIA

Query URL: http://69.55.49.190/

Started at: Thu Jun 26 2014, 05:44:56 -04:00

Finished at: Thu Jun 26 2014, 05:44:56 -04:00

ANALYSIS

This rush generated **11,633** successful hits in **60 seconds** and we transferred **790.92 MB** of data in and out of your app. The average hit rate of **194/second** translates to about **16,751,520** hits/day.

The average response time was **193 ms**.

RESPONSE TIMES	TEST CONFIGURATION	OTHER STATS
FASTEST: 37 MS	REGION: VIRGINIA	AVG. HITS: 194 /SEC
SLOWEST: 529 MS	DURATION: 60 SECONDS	DATA TRANSFERED:
AVERAGE: 193 MS	LOAD: 1-500 USERS	790.92 MB

HITS 99.99% (11633)

ERRORS 0.00% (0)

TIMEOUTS 0.01% (1)

HITS

This rush generated **11,633** 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	11265
502	HTTP	Bad Gateway	368

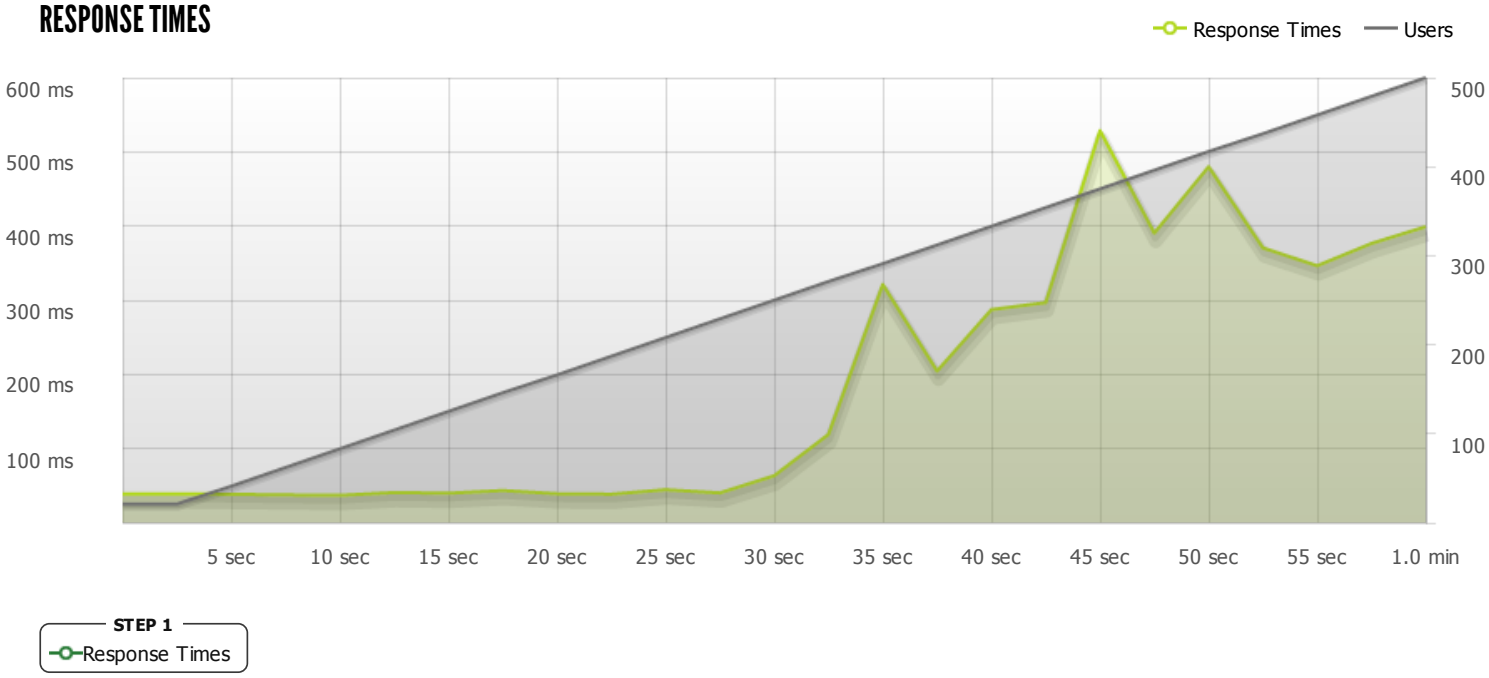
HTTP 200 OK **97%** (11265)

HTTP 502 BAD GATEWAY **3%** (368)

TIMEOUTS

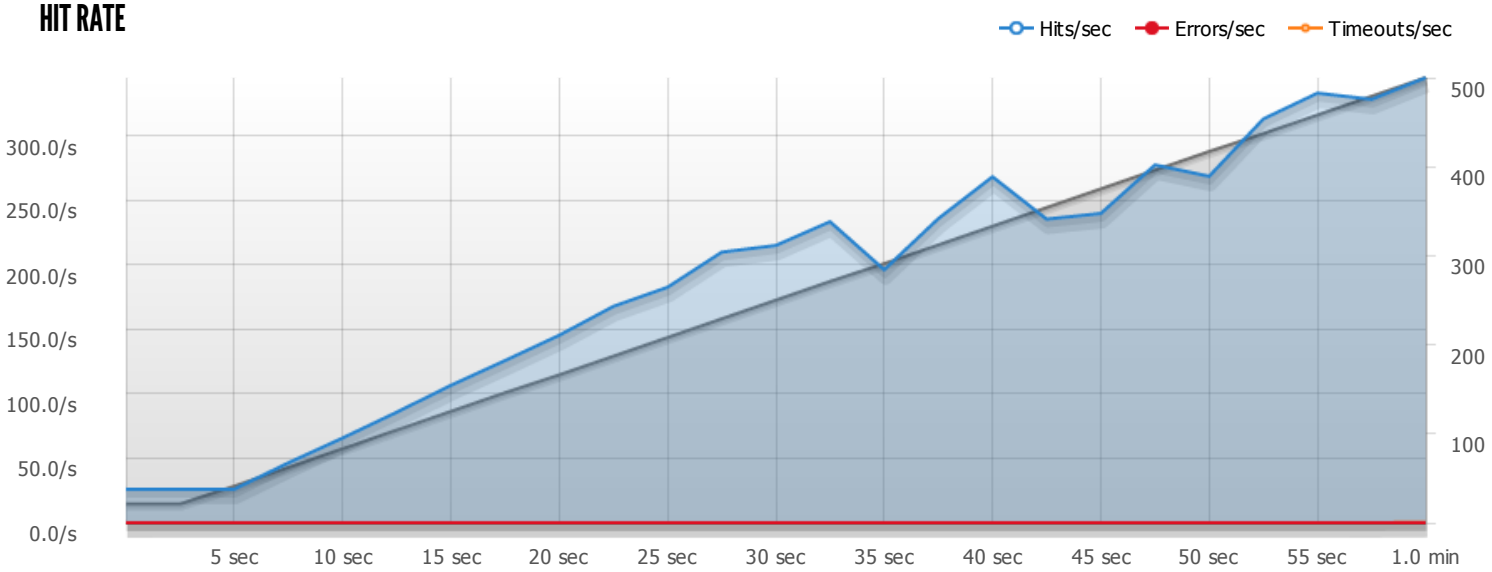
The first timeout happened at **60 seconds** into the test when the number of concurrent users was at **500**. Looks like you've been rushing with a timeout of **1000 ms**. Timeouts tend to increase with concurrency if you have lock contention of sorts. You might want to think about in-memory caching using redis, memcached or varnish to return stale data for a period of time and asynchronously refresh this data.

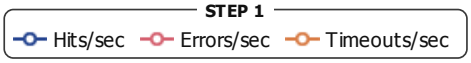
RESPONSE TIMES



The max response time was: **528 ms @ 375 users**

HIT RATE





The max hit rate was: **345 hits per second**
