Load Test Report

Date: 2/9/2017

Test from: ireland

Query URL: http://188.166.157.159/

Started at: Thu Feb 9 2017, 03:17:00 +00:00

Finished at: Thu Feb 9 2017, 03:18:00 +00:00

Test link: https://www.blitz.io/to#/play/result/ireland:d3fb5a18e42027a9d988ad2895e9821e

Analysis

This rush generated 525 successful hits in 60 seconds and we transferred 26.31 MB of data in and out of your app. The average hit rate of 9/second translates to about 756,000 hits/day.

The average response time was 422 ms.

You've got bigger problems, though: 71.67% of the users during this rush experienced timeouts or errors!

Response Times

- Fastest: 86 ms
- Slowest: 1,022 ms
- Average: 422 ms

Test Configuration

- Region: ireland
- Duration: 60 seconds
- Load: 1-100 users
- Avg. Hits: 9/sec
- Transferred: 0.24MB
- Received: 26.06MB

Other Stats

- Hits: 28.33% (525)
- Errors: 25.31% (469)
- Timeouts: 46.36% (859)

Hits

This rush generated 525 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.

<table>
<thead>
<tr>
<th>Code</th>
<th>Type</th>
<th>Description</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>HTTP</td>
<td>OK</td>
<td>525</td>
</tr>
</tbody>
</table>

Errors

The first error happened at 47.5 seconds into the test when the number of concurrent users was at 80. 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).

<table>
<thead>
<tr>
<th>Code</th>
<th>Type</th>
<th>Description</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>502</td>
<td>HTTP</td>
<td>Bad Gateway</td>
<td>469</td>
</tr>
</tbody>
</table>

Timeouts

The first timeout happened at 27.5 seconds into the test when the number of concurrent users was at 46. 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: **1021 ms @ 59 users**

Hit Rate

The max hit rate was: **24 hits per second**