Load Test Report

**Date:** 5/27/2015

**Test from:** Virginia

**Query URL:** wordpress-1055-180-371.cloudwaysapps.com

**Started at:** Wed May 27 2015, 02:40:53 -04:00

**Finished at:** Wed May 27 2015, 02:41:53 -04:00

**Test link:** https://www.blitz.io/to#/play

**Analysis**

This rush generated **52,124** successful hits in **60 seconds** and we transferred **741.57 MB** of data in and out of your app. The average hit rate of **869/second** translates to about **75,058,560** hits/day.

The average response time was **35 ms**.

You've got bigger problems, though: **5.55%** of the users during this rush experienced timeouts or errors!

**Response Times**

- Fastest: **23 ms**
- Slowest: **87 ms**
- Average: **35 ms**

**Test Configuration**

- Region: Virginia
- Duration: 60 seconds
- Load: 1-2000 users

**Other Stats**

- Avg. Hits: **869/second**
- Transferred: **8.43MB**
- Received: **733.14MB**

**Hits**

This rush generated **52,124** 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>52124</td>
</tr>
</tbody>
</table>

**Errors**

The first error happened at **20 seconds** into the test when the number of concurrent users was at **665**. 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>23</td>
<td>TCP</td>
<td>Connection timeout</td>
<td>422</td>
</tr>
<tr>
<td>500</td>
<td>HTTP</td>
<td>Internal Server Error</td>
<td>1143</td>
</tr>
</tbody>
</table>

**Timeouts**

The first timeout happened at **22.5 seconds** into the test when the number of concurrent users was at **748**. 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: **87 ms @ 1666 users**

Hit Rate

The max hit rate was: **1,672 hits per second**