**Report about conducted load test**

**Date:** 04.08.2022

**Author:** Shushanik Tonoyan

**Application:** BlogEngine.NET version 3.2

**Environment:** Test Environment

**Test Environment configuration (RAM, CPU etc.):**

|  |  |
| --- | --- |
| Processor | Intel(R) Core(TM) i7-10610U CPU 1.80 GHz 2.30 GHz |
| RAM | 3.95 GB |
| System type | 64-bit operating system |
| Operating System | Windows 10 Enterprise 21H1 |

1. **Why such testing was conducted:**
2. Performed volume test of application under test. The Load test has been conducted for different number of the text posts to determine the threshold for degradation depending on the posts count.
3. Performed load test of application under test for different types of data, namely pure text posts and text posts with media information. The results comparison of test results for different kinds of data has been conducted.
4. **Test script description:**

The following script should be run for

1. Anonymous script with probability usage is implemented according to the following table

|  |  |
| --- | --- |
| **Flow** | **Execution percentage %** |
| Home Page | 15 |
| Open Random Date | 10 |
| Open Predefined Date | 30 |
| Search by Name | 30 |
| Open Large Calendar | 10 |
| Open Contacts | 5 |
| Open Random page (yes/no) | 50/50 |
| Open post (yes/no) | 80/20 |
| Random or First | 65/35 |
| Comment (yes/no) | 20/80 |
|  |  |

1. Admin script
2. Editor script

**Anonymous script**

Diagram

Description automatically generatedDiagram

Description automatically generatedDiagram

Description automatically generated

**Admin Script**

Diagram

Description automatically generated

**Editor Script**

Diagram

Description automatically generated

1. **Tests:**   
   **Metrics to monitor:** average throughput, response time 95th percentile, error rate, and CPU usage.  
   **Test run preconditions:** 2 admins, 2 editors should be created. Volume of data according to the table above were prepared different runs:

|  |  |
| --- | --- |
| **Run number** | **Posts count** |
| **1** | **100** |
| **2** | **1000** |
| **3** | **2000** |
| **4** | **5000** |
| **5** | **2000** (1000 text & 1000 with 1 MB image) |

**Base configuration:** CPU count 1, RAM 4 GB. For the configuration mentioned, the capacity test has been conducted. According to the capacity test, the saturation point is around 95 users, and the application stays in the comfort zone while the users count is less than 75 users. On the base of this results, the average load was defined to be 40 users.

**Load Model:**

The load test for the application under test was conducted. On the base of the capacity test the average load was defined.

**Average load**: as 40 virtual users.

Load testing. Test was conducted overall for 40 users, duration 900 sec, constant delay between requests 2 sec with deviation 0.10 sec.

|  |  |  |
| --- | --- | --- |
| **Users** | **Threads count** | **Rampup in seconds** |
| Admins | 2 | 120 |
| Editors | 4 | 120 |
| Anonymous Users | 34 | 120 |

1. **Short summary on conducted tests:**

The aim of this test was to analyze the system performance by increasing the volume of test data. Detailed test results presented on the section 6 of the report. According to test results increase of the volume of the test data until 5000 text posts does not lead to the significant degradation of the system under the test.

Thereby, in terms of posts count 5000 text posts are kind of threshold. In case of 5000 posts, we observe increase of the error rate and significant degradation of other KPIs.

In the second part of the test the influence of the different types of data on the application performance has been considered. Compared test results for 2000 pure text posts and 2000 mixed posts (1000 pure text and 1000 text and 1 MB attached image). On the base of the tests results it was found that the presence of heavy data (namely images) leads to the degradation of performance. So, to determine the threshold of the application depending on test data volume, we need to consider different types of data.

1. **Detailed test results:**

**Volume test results**



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Text posts count | 100 | 1000 | 2000 | 5000 |
| Throughput (Req/s) | 18.3 | 16 | 15.5 | 13.1 |
| Response Time 95th percentile | 157 ms | 283 ms | 288 ms | 666 ms |
| Error Rate % | 0.02 | 0.03 | 0.11 | 0.38 |
| CPU usage % | 31.1 | 40.7 | 53.2 | 62.2 |

**1 Test was conducted for 100 text posts created.**

**Chart

Description automatically generated**

**A screenshot of a computer

Description automatically generated with medium confidence**

**A screenshot of a computer

Description automatically generated**

**Graphical user interface, text

Description automatically generated**

From the charts above, one can see, that for the initial data set the behavior of application is stable, the average CPU usage is about 31% and RAM usage is about 84 %.

**2 Test was conducted for 1000 text posts created.**

**Chart, histogram

Description automatically generated**

**Chart, histogram

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

Graphical user interface, text

Description automatically generated

After increasing created text posts count 10 times, we can observe significant degradation of KPIs, particularly, the average throughput decreased by ~13%, average response time increased by 80%, CPU usage increased by ~31%. Nevertheless, the behavior of the application is stable2, no errors are observed.

**3 Test was conducted for 2000 text posts created.**

**Chart

Description automatically generated**

**Chart, histogram

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Graphical user interface, text

Description automatically generated**

The further duplication of text post count leads just a slight degradation of the KPIs: the average throughput decreased by ~3%, average response time increased by 1.7%, but CPU usage increased again by ~31%.

**4 Test was conducted for 5000 text posts created.**

**Chart

Description automatically generated**

**Chart, histogram

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Graphical user interface, text

Description automatically generated**

If we have 5000 pure text posts created the application degradants significantly. We can observe that error rate increased dramatically.

Graphical user interface, text

Description automatically generated

The same is observable for the average response time: increased by 131%. On the application response time chart a spike is observable. The response time reached the value 1.71 s.

**Test with different types of data**

For understanding the behavior of application in case of different types of data, the following test data have been prepared:

1000 pure text posts

1000 posts with a text and attached 1MB image

Chart, histogram

Description automatically generated

Chart

Description automatically generated

A screenshot of a computer

Description automatically generated

Graphical user interface, text

Description automatically generated

|  |  |  |
| --- | --- | --- |
| Text posts count | 2000 | 1000 |
| Posts count with attached image | 0 | 1000 |
| Throughput (Req/s) | 15.5 | 15.3 |
| Response Time 95th percentile | 288 ms | 390 ms |
| Error Rate % | 0.11 | 0.14 |
| CPU usage % | 53.2 | 60.9 |

From the comparison of results it follows, that despite the posts total count is the same for both runs results are quite different. Presence of the posts containing images leads to the degradation of applications performance. Particularly, we observe increase of the response time by more than 35%, error rate increased by 27%, CPU increased by 14%. This degradation should be considered when determining the threshold for degradation depending on the quantity of the posts. It depends not only on the posts count but also on posts type (text or mixed).

1. **Conclusion: On the base of the conducted load tests result it has been shown that in case of the pure text posts, the threshold is 5000 posts. Starting from that data volume the degradation of system performance is observed. The error rate significantly getting higher, average response increases by almost two and a half times time. A spike is observable on the response time chart. Investigations show that presence of a large volume of data leads to the CPU usage increase, sometimes up to 100%, which in turn causes slowness in requests processing. Also, it has been shown that performance of the application depends not only on posts count but also on posts type. The presence of attachments in the post’s degradants the application performance.**