**Report about conducted scalability test**

**Date:** 1-3 August 2022

**Author:** Oleksandr Maksymenko

**Application:** BlogEngine.NET version 3.2

**Environment:** Host machine where script was run + Virtual Machine where blog is deployed

1. **Why such testing was conducted:** To find threshold for degradation depending on the quantity of the posts.
2. **Test script description:** Current script is combined script of 3 different role user scenarios: Anonymous, Admin and Editor.
   1. **Anonymous Script:** contains different user scenarios which are chosen with different probabilities (Open Home Page, Open Random Date, Open Predefined Date, Search Post by Name, Open large calendar, Open contacts). First three scenarios are opening random page with posts (in 50% cases). All scenarios (except Open Contacts) have Open Page script, which is used randomly in ~80% cases. Detailed description can be found on screenshots and in tables below.
   2. **Admin Script:** contains actions for Admin flow. Firstly “user” opens Home page, Logs In, after that starts loop which executes 10 times: Open Admin Page -> Open Users Page -> depending on current number of users -> Create User OR Delete User. After loop ends “user” will be logged off.
   3. **Editor Script:** contains actions for Editor flow. Firstly “user” opens Home page, Logs In, after that starts loop which executes 50 times: Open Predefined Date -> Open Random Page (in case editable post was not found on first page) -> Open Random Post -> Edit Post. After loop ends “user” will be logged off.

**Diagram

Description automatically generatedDiagram

Description automatically generatedDiagram

Description automatically generated**

Diagram

Description automatically generatedDiagram

Description automatically generated

# Tests preconditions and configurations

**Test run preconditions:**

* 100, 1000, 2000 or 5000 pregenerated posts on predefined dates
* CSV file with 10 predefined dates
* CSV file with Admin credentials
* CSV file with Editor credentials
* 1000 pregenerated media posts

**Monitored Key Performance Indicators:**

* Throughput (transactions per second)
* Error rate, %
* Average response time, ms
* Median response time, ms
* 90% response time, ms
* 95% response time, ms

**Why that Load Model was chosen:** task 7 test execution showed saturation point on ~90 users, so number of anonymous users was set to 60 users, editor users to 8, and admin users remained the same at 2. Overall number of users equals to 70 in that load test, which is close to saturation point, but at the same time system does not work on 100% CPU load all the time (from Detailed test results we can see average CPU load was 76% for that config), which means we should have stable processing time during execution. Rampup period was set to 2 minutes, and other 8 minutes we had stable load.

**Load Model:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Users** | **Ramp up (s)** | **Duration (s)** |
| **Anonymous** Users | 60 | 120 | 600 |
| **Admin** Users | 2 |
| **Editor** Users | 8 |

**Comparison of Anonymous script step probabilities:**

|  |  |
| --- | --- |
| **Action** | **Probability** |
| Open Home Page | 15% |
| Open Random Date | 10% |
| Open Predefined Date | 30% |
| Search by Name | 30% |
| Open Large Calendar | 10% |
| Open Contacts | 5% |

|  |  |
| --- | --- |
| **Action:** Open Random Page | **Probability** |
| Yes | 50% |
| No | 50% |

|  |  |
| --- | --- |
| **Action:** Open Post | **Probability** |
| Yes | 80% |
| No | 20% |

|  |  |
| --- | --- |
| **Action:** Open First or Random Post | **Probability** |
| First | 35% |
| Random | 65% |

|  |  |
| --- | --- |
| **Action:** Send Comment | **Probability** |
| Yes | 20% |
| No | 80% |

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

|  |  |
| --- | --- |
| **RAM** | 4096 MB |
| **CPU** | 1 Core CPU (Intel(R) Core(TM) i7-8665U CPU @ 1.90GHz 2.11 GHz) |
| **System Type** | Windows 10 64-bit |

# Short summary on conducted tests:

**Test with different number of the text posts:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Number of Posts** | 100 | 1000 | 2000 | 5000 |
| **Throughput (transactions per sec)** | 22.09 | **22.11** | 21.05 | 15.81 |
| **Error rate, %** | **0** | 0.01 | 0.68 | 1.29 |
| **Average response time, ms** | 182.26 | **170.76** | 345.43 | 1413.97 |
| **Median response time, ms** | 126.5 | **122.5** | 621.5 | 3010.5 |
| **90%, ms** | 1210 | **1159** | 1955.6 | 5070.5 |
| **95%, ms** | **1704.35** | 1913.4 | 2578.95 | 5747.65 |

# Detailed test results:

## Text posts only

### 100 text posts

**Composite Graph (Active Threads, Throughput, Average Response Time, Errors per sec)**

Chart, histogram

Description automatically generated

**Aggregate Report**

Calendar

Description automatically generated with low confidence

**All transaction response times**

Chart, histogram

Description automatically generated

**CPU load**

Chart

Description automatically generated with medium confidence

**Memory usage**

A screenshot of a video game

Description automatically generated

**Garbage Collector**

A screenshot of a computer

Description automatically generated with medium confidence

**Cache**

A screenshot of a computer

Description automatically generated with medium confidence

### 1000 text posts

**Composite Graph (Active Threads, Throughput, Average Response Time, Errors per sec)**

Chart

Description automatically generated

**Aggregate Report**

A picture containing text, screen

Description automatically generated

**All transaction response times**

Chart, histogram

Description automatically generated

**CPU load**

A screen shot of a video game

Description automatically generated with medium confidence

**Memory usage**

A screenshot of a video game

Description automatically generated

**Garbage Collector**

A screenshot of a computer

Description automatically generated with medium confidence

**Cache**

A screenshot of a computer

Description automatically generated with medium confidence

### 2000 text posts

**Composite Graph (Active Threads, Throughput, Average Response Time, Errors per sec)**

Graphical user interface, chart

Description automatically generated

**Aggregate Report**

A picture containing text, computer

Description automatically generated

**All transaction response times**

Chart, histogram

Description automatically generated

**CPU load**

A picture containing text, night, dark

Description automatically generated

**Memory usage**

A screenshot of a computer

Description automatically generated with medium confidence

**Garbage Collector**

A screenshot of a computer

Description automatically generated with medium confidence

**Cache**

A screenshot of a computer

Description automatically generated with medium confidence

### 5000 text posts

**Composite Graph (Active Threads, Throughput, Average Response Time, Errors per sec)**

Chart, histogram

Description automatically generated

**Aggregate Report**

A picture containing text, outdoor, computer

Description automatically generated

**All transaction response times**

Chart

Description automatically generated

**Errors**

Graphical user interface

Description automatically generated

**CPU load**

A screenshot of a video game

Description automatically generated

**Memory usage**

A screenshot of a video game

Description automatically generated

**Garbage Collector**

A screenshot of a computer

Description automatically generated with medium confidence

**Cache**

A screenshot of a computer

Description automatically generated with medium confidence

## Mixed posts

### 1000 text + 1000 media posts

**Composite Graph (Active Threads, Throughput, Average Response Time, Errors per sec)**

**Aggregate Report**

**All transaction response times**

**CPU load**

**Memory usage**

**Garbage Collector**

**Cache**

# Conclusion:

* We executed test for 8 different configuration which are listed in “Tests preconditions and configurations” section.
* First 4 runs were executed for case with fixed RAM 8 GB and scalable CPU: they showed most optimal CPU case is 2 Cores, because of lowest error rate and all response time metrics. Important to note that 3 Cores CPU also showed good results which were close to 2 Core case, but still they were bit worse than 2 Core. If we include here cost of purchase and maintain – 2 Core CPU is most optimal to run BlogEngine application.
* Next 3 runs were executed for fixed CPU 2 Cores and scalable RAM: they showed some unstable results for 2 GB and 6 GB cases, while 4 GB case showed better results in response time comparing to 8 GB which we executed in previous part.
* Comparing of Default config (1 Core CPU + 4 GB RAM) and the most optimal after scaling of RAM and CPU (2 Core CPU + 4 GB RAM):

|  |  |  |
| --- | --- | --- |
| **CPU** | 1 Core | 2 Core |
| **RAM** | 4 GB | 4 GB |
| **Throughput (transactions per sec)** | 21.44 | **22.54** |
| **Error rate, %** | **0.03** | 0.11 |
| **Average response time, ms** | 255.62 | **98.65** |
| **Median response time, ms** | 205.5 | **72** |
| **90%, ms** | 1191.2 | **175.8** |
| **95%, ms** | 2235.8 | **578.1** |

* Looking into comparing of these two cases we can **conclude** that additional 0.08% error rate and slight increase of costs between 1 Core CPU and 2 Core CPU can be acceptable to have **65% lower median** response time, **85% lower 90 percentile** response time and **74% lower 95** **percentile** response time. So, we choose **2 Core CPU and 4 GB RAM** as most optimal configuration to run BlogEngine application.