**Report about conducted load test**

**Date:** 22nd of May 2022

**Author:** Oleksandr Maksymenko

**Application:** BlogEngine.NET version 3.2

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

**Test 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 |

1. **Why such testing was conducted:** To determine general capacity of BlogEngine application, find saturation point
2. **Test script description:** Script contains different user scenarios which are chosen randomly (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 ~50% cases. Detailed description can be found on screenshots.

**Diagram

Description automatically generatedDiagram

Description automatically generatedDiagram

Description automatically generated**

1. **Tests:** 2 test runs.  
     
   **Test run preconditions:**

* 100 pregenerated posts on predefined dates
* CSV file with 10 predefined dates

**Load Model 1:**

|  |  |
| --- | --- |
| **Users** | 60 |
| **Rump up time (s)** | 600 |
| **Duration (s)** | 600 |

**Load Model 2:**

|  |  |
| --- | --- |
| **Users** | 80 |
| **Rump up time** | 600 |
| **Duration** | 600 |

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

* in test execution #1 we can see very high response time at the beginning of execution, but after slow start AVG response time was almost permanent during whole run, with some spike at the end. Total throughput was constantly growing, until some errors appeared after 57th thread started its execution.
* In test execution #2 number of active threads were increased from 60 to 80 users. Response time was not so high at the start but huge spike can be observed during 300-410 seconds of run, total throughput was growing even after errors were started to happen but little drop can be observed at 300th second of run, when response time when higher, at that time 39-44 active threads. A lot of errors started after 59th user became active.

1. **Detailed test results:**

**5a) Test run 1 with load model 1**

**Composite Graph** (Active Threads, Throughput, AVG response time)

**A screenshot of a computer

Description automatically generated with medium confidence**

**A screenshot of a computer

Description automatically generated with medium confidence**

Max response time started growing at the point of 15-17 active users

**Chart

Description automatically generated**

**5b) Test run 2 with load model 2**

Some data is missed during 290-300 seconds of execution

**Composite Graph** (Active Threads, Throughput, AVG response time)

Graphical user interface

Description automatically generated

**A screenshot of a computer

Description automatically generated with medium confidence**

Response time has huge spike when application had 40-55 active users

A screenshot of a video game

Description automatically generated

1. **Conclusion:** After 2 test runs we can conclude our system is stable before it reaches 57-59 concurrent users, at that moment they produce 18-20 requests per second.

**UPD from 28/05/2022:** Updated test run with ignored embedded resources errors and more open ports

**Load Model 2:**

|  |  |
| --- | --- |
| **Users** | 500 |
| **Rump up time** | 600 |
| **Duration** | 600 |

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

Chart, histogram

Description automatically generated

A screenshot of a video game

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

System reached saturation point at ~110 users and 25-27 transactions per second.

Break point was reached at 360-370 concurrent users.

More details will be covered in report for task 6.