**Report about conducted load test**

**Date:** 15th of June 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, compare system behavior for different user scenarios
2. **Test script 6 description:** 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.

**Diagram

Description automatically generatedDiagram

Description automatically generatedDiagram

Description automatically generated**

**Comparison of script step probabilities:**

|  |  |  |
| --- | --- | --- |
| **Action** | **Task 3 script** | **Task 6 script** |
| Open Home Page | ~16,7% | 15% |
| Open Random Date | ~16,7% | 10% |
| Open Predefined Date | ~16,7% | 30% |
| Search by Name | ~16,7% | 30% |
| Open Large Calendar | ~16,7% | 10% |
| Open Contacts | ~16,7% | 5% |

|  |  |  |
| --- | --- | --- |
| **Action:** Open Random Page | **Task 3 script** | **Task 6 script** |
| Yes | 50% | 50% |
| No | 50% | 50% |

|  |  |  |
| --- | --- | --- |
| **Action:** Open Post | **Task 3 script** | **Task 6 script** |
| Yes | 50% | **80**% |
| No | 50% | 20% |

|  |  |  |
| --- | --- | --- |
| **Action:** Open First or Random Post | **Task 3 script** | **Task 6 script** |
| First | 50% | 35% |
| Random | 50% | **65**% |

|  |  |  |
| --- | --- | --- |
| **Action:** Send Comment | **Task 3 script** | **Task 6 script** |
| Yes | 50% | 20% |
| No | 50% | **80**% |

1. **Tests:**

* 1 run of Task 3 script with 100 pregenerated posts
* 1 run of Task 3 script with 1000 pregenerated posts
* 1 run of Task 6 script with 100 pregenerated posts
* 1 run of Task 6 script with 1000 pregenerated posts

**Test run preconditions:**

* 100 or 1000 pregenerated posts on predefined dates
* CSV file with 10 predefined dates

**Load Model:**

|  |  |
| --- | --- |
| **Users** | 500 |
| **Rump up time (s)** | 600 |
| **Duration (s)** | 600 |

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

* 100 posts executions

|  |  |  |
| --- | --- | --- |
|  | **Task 3 script** | **Task 6 script** |
| **Saturation Point** | 95-97 users | 165-170 users |
| **Break Point** | 355 users | 345 users |
| **Throughput** | 26 req/s | 39,7 req/s |

Overall, Blog performance is better in case with different probabilities for key user transactions, as Saturation Point and Total Throughput are higher in that case. Exception is Break Point, which is nearly the same for both scripts.

* 1000 posts executions

|  |  |  |
| --- | --- | --- |
|  | **Task 3 script** | **Task 6 script** |
| **Saturation Point** | 115-117 users | 130-140 users |
| **Break Point** | 390 users | 400 users |
| **Throughput** | 34,8 req/s | 34,5 req/s |

In case of 1000 posts behavior of both test scripts is similar to each other, except Saturation Point which is still lower in case of task3 script. Also, it is important to say that task3 script with same probabilities for user transactions works better with high number of posts.

* Task 6 executions

|  |  |  |
| --- | --- | --- |
|  | **100 posts** | **1000 posts** |
| **Saturation Point** | 165-170 users | 130-140 users |
| **Break Point** | 345 users | 400 users |
| **Throughput** | 39,7 req/s | 34,5 req/s |

Different probabilities scenario is behaving worse for all key application metrics (except Break Point), which can signalize us that task6 script is more optimized for usage with low number of posts in Blog.

1. **Detailed test results: 100 posts comparison (task3 / task6)**
2. 100 Posts: Active Threads, Throughput, Errors

Chart, line chart

Description automatically generated

Chart, line chart

Description automatically generated

We can observe Active Threads was growing linearly up to 500 in both graphs, higher average throughput for task6 script (~40, while it was ~30 in task3 script), number of errors started to grow approximately at the same time for both scripts.

1. 100 Posts: Composite Graph

A screenshot of a video game

Description automatically generated

Chart

Description automatically generated

1. 100 Posts: AVG Response Time for each transaction

Chart

Description automatically generated

Chart

Description automatically generated

AVG response time started to grow up after Saturation Point, had its peak at ~15s for all transactions in task3 and ~7s in task6. After reaching of Break Point we can observe slight drop of response time in task3, but in task6 it remained the same.

1. 100 Posts: Aggregate Report (Error% column to be fixed in further tasks)

A picture containing calendar

Description automatically generated

A screen shot of a computer

Description automatically generated with low confidence

General number of sent transactions in task6 is 53% higher than in task3; Response time is less for each transaction in task6; Error rate dropped from 14% in task3 to 5% in task6.

1. 100 Posts: Errors

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

1. 100 Posts: CPU, Memory, Network

A screenshot of a computer

Description automatically generated with medium confidence

Chart

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a video game

Description automatically generated

1. **Detailed test results: 1000 posts comparison (task3 / task6)**
2. 1000 Posts: Active Threads, Throughput, Errors

Graphical user interface, chart, line chart

Description automatically generated

Graphical user interface, chart, line chart

Description automatically generated

We can observe Active Threads was growing linearly up to 500 users in both graphs, throughput graph is similar to each other with larger instability in task6 script, number of errors started to grow approximately at the same time for both scripts and it was larger in task6 script.

1. 1000 Posts: Composite Graph

Graphical user interface, chart

Description automatically generatedChart

Description automatically generated

1. 1000 Posts: AVG Response Time for each transaction

Chart, histogram

Description automatically generated

Chart, histogram

Description automatically generated

AVG response time started to grow up after Saturation Point, had its peak at ~10-12s for most of transactions both tasks. After reaching of Break Point we can observe slight drop of response time in task3, but in task6 it remained the same.

1. 1000 Posts: Aggregate Report (Error% column to be fixed in further tasks)

A screen shot of a computer

Description automatically generated with low confidence

Calendar

Description automatically generated with medium confidence

All metrics are nearly the same for both test scripts with 1-2% lower numbers in task6 script.

1. 1000 Posts: Errors

Graphical user interface, text, application

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

1. 1000 Posts: CPU, Memory, Network

Chart

Description automatically generated

Chart, histogram

Description automatically generated

Graphical user interface

Description automatically generated

A screenshot of a video game

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

1. **Detailed test results: Task 6 comparison (100 posts / 1000 posts)**
2. Task 6: Active Threads, Throughput, Errors

Chart, line chart

Description automatically generated

Graphical user interface, chart, line chart

Description automatically generated

We can observe Active Threads was growing linearly up to 500 users in both graphs, average throughput was higher in run with 100 posts, number of errors started to grow approximately at the same time for both scripts.

1. Task 6: Composite Graph

Chart

Description automatically generated

Chart

Description automatically generated

1. Task 6: AVG Response Time for each transaction

Chart

Description automatically generated

Chart, histogram

Description automatically generated

AVG response time started to grow up after Saturation Point, had its peak at ~7s for 100 posts and ~9-11s for 1000 posts.

1. Task 6: Aggregate Report (Error% column to be fixed in further tasks)

A screen shot of a computer

Description automatically generated with low confidence

Calendar

Description automatically generated with medium confidence

General number of sent transactions for 1000 posts is 13% lower than in case of 100 posts; Response time in 1000 posts case is ~30% higher for most of transactions except Open Main Page (4.4% higher for 1000 posts); Error rate raised from 5% for 100 posts to 8% for 1000 posts.

1. Task 6: Errors

Graphical user interface, text, application

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

1. Task 6: CPU, Memory, Network

Chart

Description automatically generated

Chart, histogram

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a video game

Description automatically generated

A screenshot of a video game

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

1. **Conclusion:**

* Task 6 script was better optimized for 100 posts run than task 3 script, it had higher throughput, general number of transactions and saturation point, while break point was nearly the same, also it had lower average response time for all of transactions.
* For test run with 1000 posts there was practically no difference between task 3 and task 6 scripts, except saturation point, which still was higher for task 6 script, but all other metrics (response time, error rate, throughput) were 1-2% lower in task 6 script.
* Significant degradation of performance can be observed in comparison of task 6 script runs for 100 and 1000 posts. Saturation point for 100 posts was 40 req/s, for 1000 posts it dropped to 35 req/s, average response time was 30% higher in case of 1000 posts, error rate increased from 5% to 8%. The exception is Break Point, which is higher in case of 1000 posts.
* Differences in server metrics (CPU, Memory, Network) can happen due to different execution time for 100 and 1000 posts runs.
* To conclude we can say that **task 3 script works better for higher number of pregenerated posts, while task 6 script can be applied for low number of pregenerated posts, more tests are required to find optimal number of posts for each test script.**