Capacity test (1000 posts vs 100 posts)

Test purpose

The purpose of the test was to find the capacity of the BlogEngine application for scenario of using the application by clients with Anonymous role. The numbers of generated blog posts for scenario are 100 and 1000.

Application Overview

BlogEngine.NET is an open source ASP.NET project that was born out of desire for a better blogging platform. Developers focused on simplicity, ease of use, extendibility and innovative design while taking advantage of the latest .NET features.

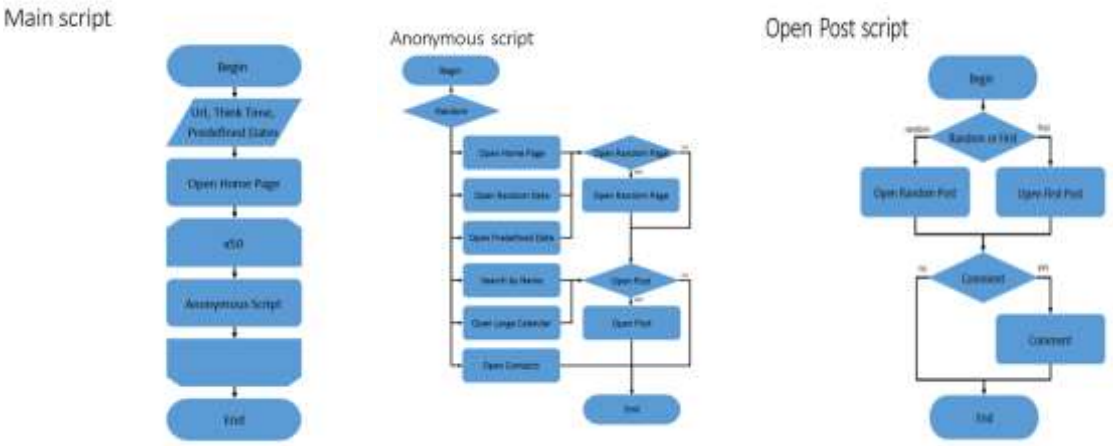
Testing Scope

Performance Testing for the following modules are in Scope of Testing

1. Home page
2. Calendar
3. Post page
4. Create comment
5. Contact page
6. Search by Name

Test scenario

There are considered Main script for Anonymous user scenario for two type of probability usage where



Test Environment & Tools

Environment: Analysis was performed on TEST environment.

General info:

	Host	Type	IP	Hosted Applications	Ports
STAGING	EPUAKIYW1844T2	VM	10.17.175.58	EPUAKIYW1844T2.kyiv.epam.com	80, 443

System resources (TEST env):

	Operational System	CPU, GHz	Memory, Gb	Disk size, Gb
DGL	Win Server 2010 R2 SP1 64bit	2	4	50

Tools:

Creating and executing tests	Apache JMeter v5.0
Storing test results and application indicators	InfluxDB 1.7.3
Collecting application metrics	Telegraf 1.10
Visualizing metrics, creating dashboards	Grafana 5.4.3

Test conditions

Condition	Transaction	Probabilities usage
1	open home page	15%
	open random date	10%
	open prediction date	30%
	search by name	30%
	open large calendar	10%
	open contacts	5%
2	open random page	50%
3	open post	80%
4	open random post	65%
	open first post	35%
5	add comment	20%

Test Setup

number of virtual users	120	120
ramp-up period	1200	1200
think time between transactions(sec)	3-5	3-5
count of posts	100	1000

Note: here random date is from 2019-02-01 to 2019-02-11.

Test Results

Test Summary

1. The maximum capacity of the application usage by Anonymous clients is not found. For both cases it is restricted by the 100% CPU usage. Adding more load leads to the application becomes unresponsive.

2. Crash point is 66 for the 100 posts case. Crash point is 96 for the 1000 posts case.

Issues found

"Comment" transaction showed 2 errors during the test. This may be caused by the concurrency during access to the same resource (post) or this transaction is implemented not correct.

More than 10 percent of "Open Home page" transaction have response time more than 23 seconds in second case.

Recommendation

Possibility to increasing CPU power. To fix problem with response time of opening Home page.

Results: graphs and tables

Test result dashboard:

First probabilities usage:
https://snapshot.raintank.io/dashboard/snapshot/0qn4EGs7qOmPA0WoRwBJHos6UoGX8nMi

Second probabilities usage:
https://snapshot.raintank.io/dashboard/snapshot/VmLTevq3riaZJZvOMdt4FApHRo1dtx1E

Host monitoring dashboard:

First probabilities usage:
https://snapshot.raintank.io/dashboard/snapshot/z1ZzR6HOhtD9vZgUSh34rL74zl80Iyf5

Second probabilities usage:
https://snapshot.raintank.io/dashboard/snapshot/n2mfx4433TsejAkquLGx4P3Q4rPruy4m

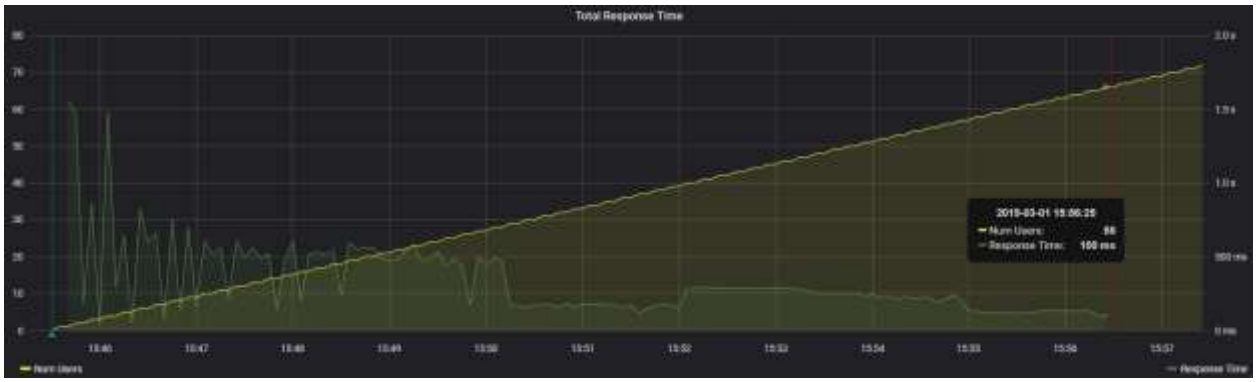
1. Aggregate Reports

100 posts:

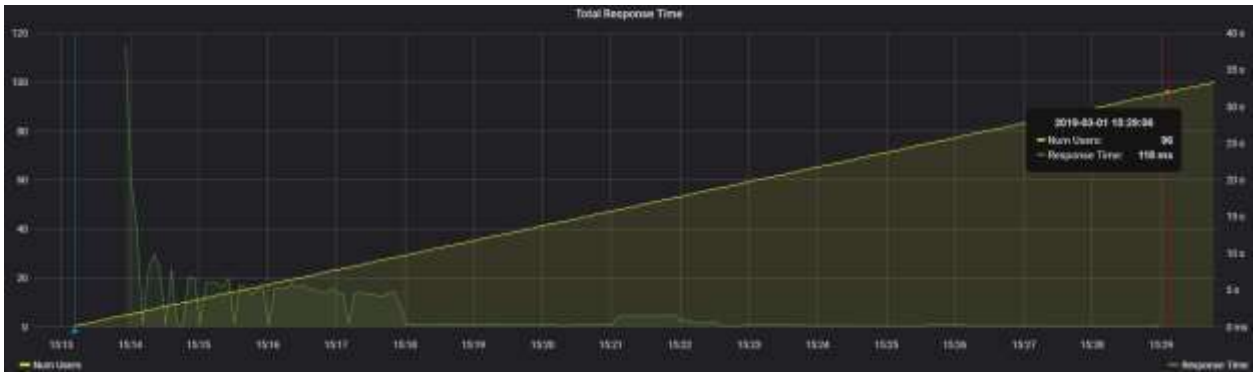
1000 posts:

2. Total Response Time vs Threads

100 posts:



1000 posts:



3. Transaction Response Times

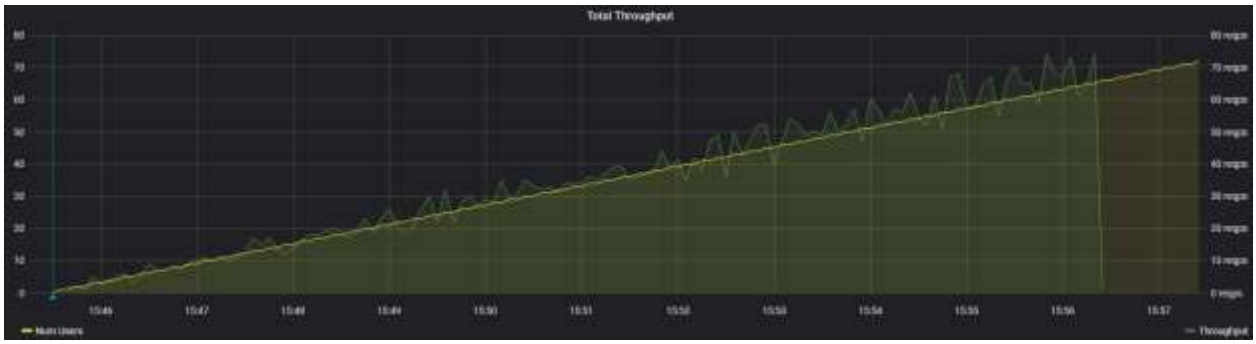
100 posts:

1000 posts:

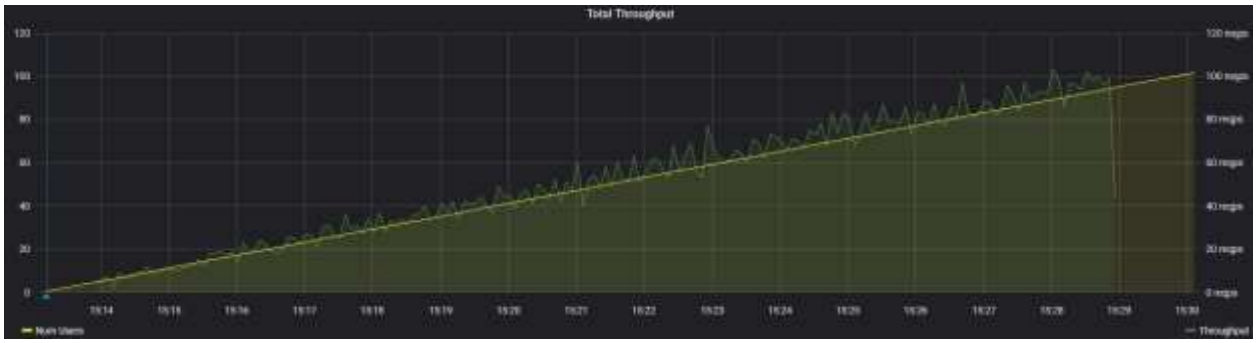
The average of response time in second case was much less than in first one.

4. Total Throughput vs Threads

100 posts:

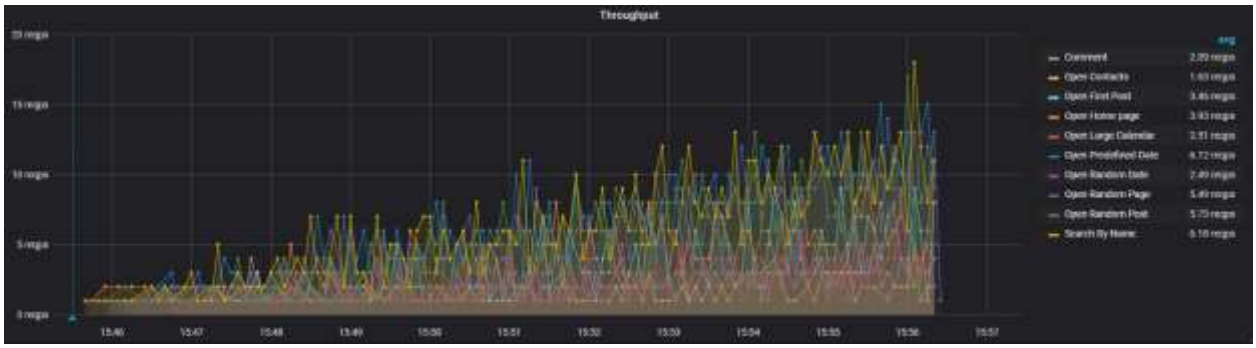


1000 posts:

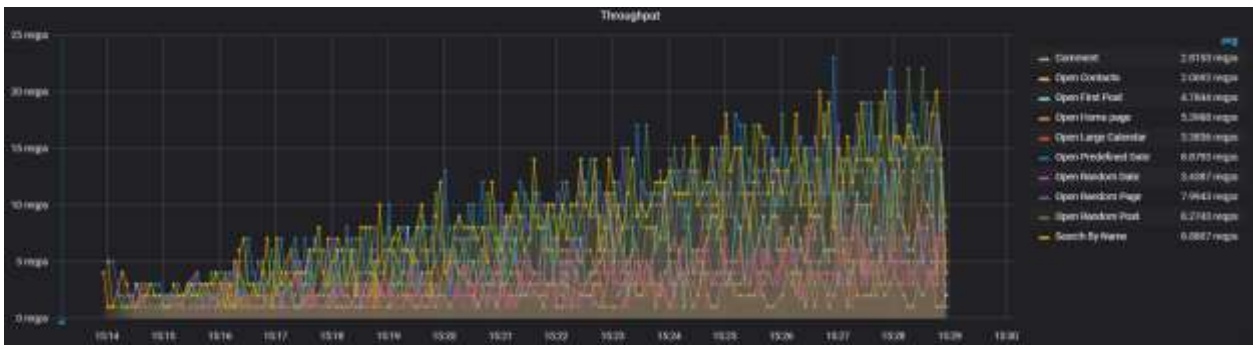


5. Transaction Throughput

100 posts:



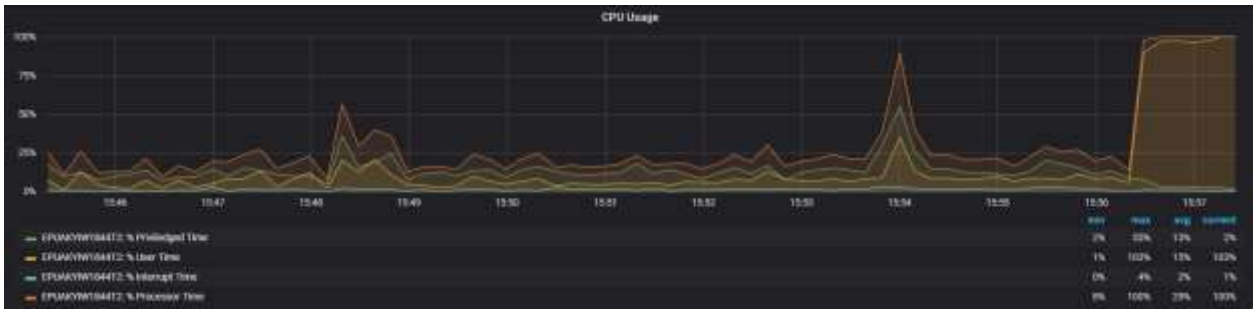
1000 posts:



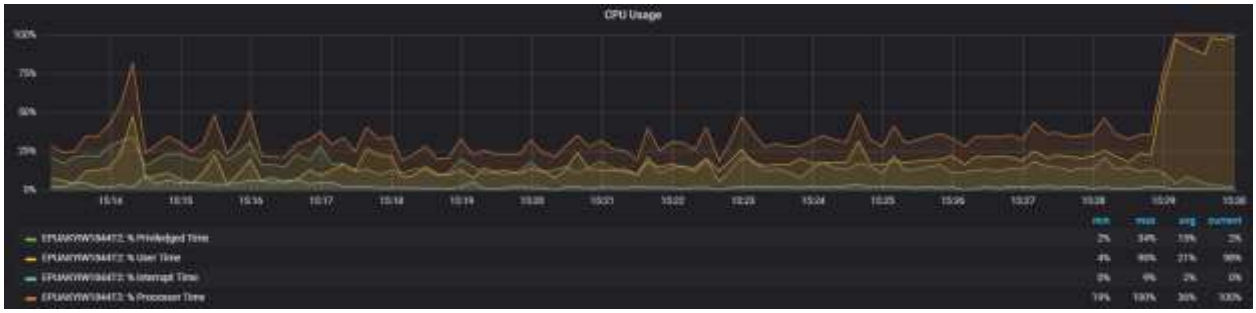
In first case throughput increased linear.

6. CPU

100 posts:



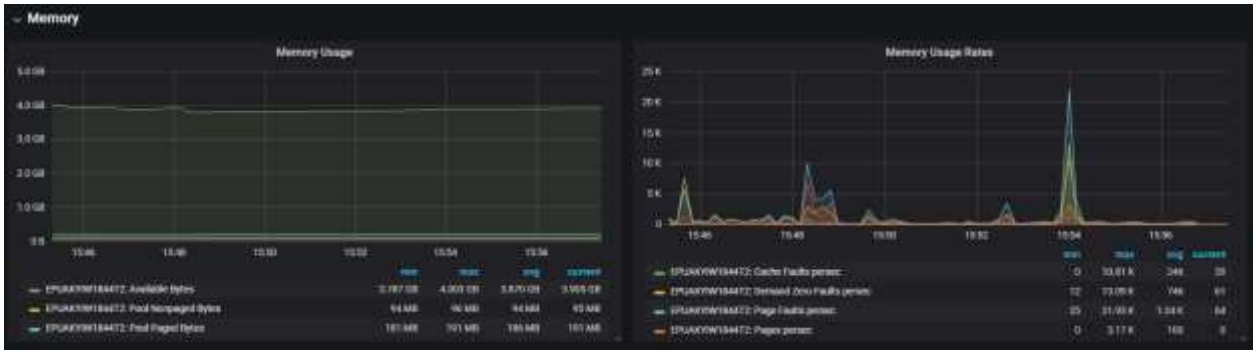
1000 posts:



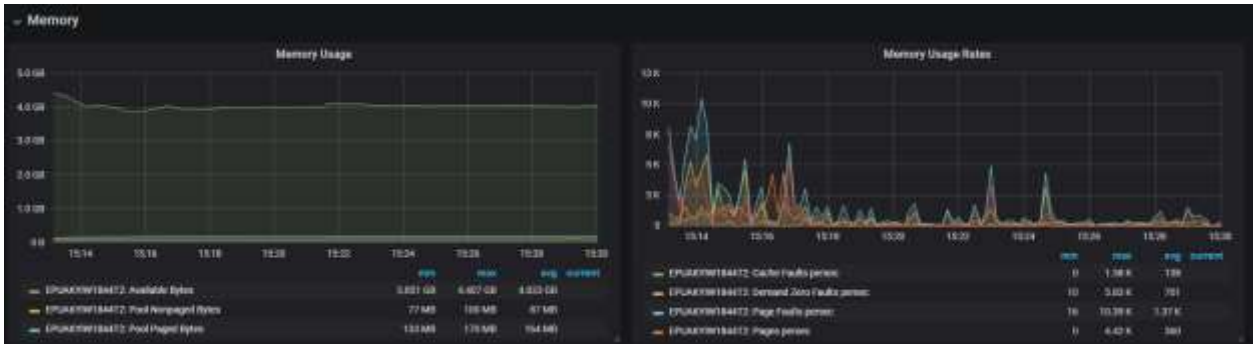
In each case indicators of CPU were stable during the test.

7. Memory

100 posts:

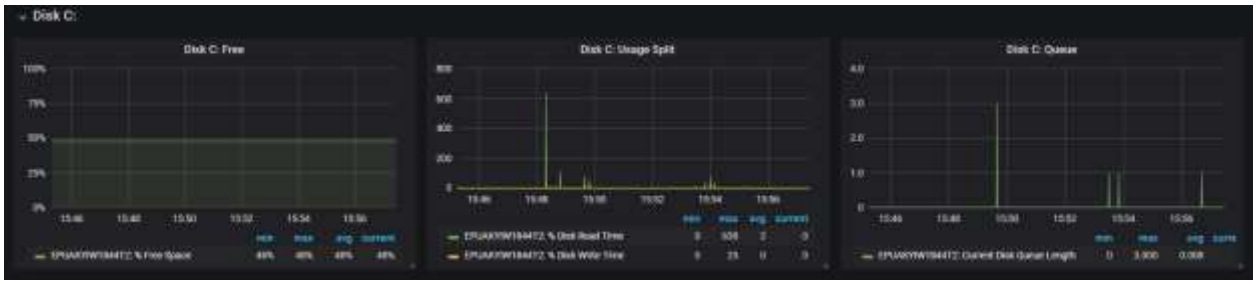


1000 posts:



8. Disk C

100 posts:



1000 posts:

9. Errors

100 posts:

1000 posts:

In second case errors appeared because probably there are mistakes in test script.

In first case no error.