Performance Testing Strategy

1.Introduction

The purpose of this document is to describe how the requirements for the BlogEngine application will be tested and verified in a test environment. The document describes the scenarios, tests, parameters and data that are created and used for this purpose. A lot of tests described here are needed to verify how the application performs its functions under high load. This information is for internal use and should not be made available to the public.

This document contains a test strategy for testing a BlogEngine application, which consists of the following items:

* Features to be tested
* Features not to be tested
* Approaches
* Requirements
* Environmental needs
* Test data
* Responsibilities
* Schedule
* Risk and contingencies
* Approvals

2. Features to be tested

* Pages with posts articles
* Post page
* Calendar
* Contacts
* Search
* Administration page
* Creating posts
* Editing posts
* Creating and deleting users
* Writing comments
* Database and File System data sources

3. Features not to be tested

* Archive
* Pages
* Categories
* Tags
* Plugins
* Changing roles

4. Approaches

4.1 Test types

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N | Test type | Obligation | Period | Notes |
| 1 | Smoke | Mandatory | Before each test |  |
| 2 | Capacity | Mandatory | After changes |  |
| 3 | Load test | Mandatory | Regular | (45%, 60% and 85% of maximum load) |
| 4 | Stress | Mandatory | Rare, after changes in capacity or like a part of preparations for high load |  |
| 5 | Scalability | Optional | Once and more after production load increases |  |
| 6 | Volume | Mandatory | As needed to define maximum volumes of data for different data sources |  |
| 7 | Stability | Mandatory | After significant changes |  |

4.1.1 Smoke Testing

Test to verify that the application is working. Perform each time after changes in the application as well as before each test.

4.1.2 Capacity Testing

Test to check the maximum capacity of the application, to check the overall performance and determine the maximum load to be handled. Should be performed after changes in the application or environment configuration.

4.1.3 Load Test

Test to verify the current state of the application and to verify that its performance meets the requirements and past versions of the application. Should be performed after each development iteration regularly.

4.1.4 Stress Testing

A test to gather information about an application under higher load than expected, its condition, performance and fault tolerance. Should be performed after changes in capacity and could be done after some changes in application or by special request.

4.1.5 Scalability Testing

Could be done to check performance multiplier(s) for different number of environment components, or changes in hardware. Also, should be performed after significant increases in production load to get ready for server components changes.

4.1.6 Volume Testing

Test to see how much performance changes when working with large volumes of data, how this affects stability. Should be done at least once and more after changes in data source configurations.

4.1.7 Stability Testing

A test to check how the application has been running for a long time, how resources are being utilized, connections and anything that might change in the application over a long period of time under load. Should be done at least once, and after significant code changes.

5. Requirements

Home page must not load more than 3 seconds. All non-functional requirements should be defined after getting results from the first round of performance testing.

6. Technical needs

6.1 Test environment

For performance testing a dedicated environment needed.

* Virtual machine emulated with Oracle VM VirtualBox Manager Version 6.0.10 with IIS installed and used as Server Service.
* SQLite 3.31.0

6.2 Testing tools

Performance testing require tools:

* Apache JMeter (5.1.1) – used for designing test scenario, generating load, gathering metrics and sending them to DB.
* InfluxData Telegraf (1.13) – used for gathering server-side metrics and sending them to DB.
* InfluxDB (1.7.8) – used to store test data for visualization.
* Grafana (6.6.2) – tool for visualizing data for monitoring and fast analysis.

7. Test Data

Requirements to test data:

1. Reusable
2. Can be access to at any time
3. Cleanable

These should be

* Posts
* Users
* Media data

8. Performance Entry, Exit, and Suspension Criteria

* 1. Entry Criteria
* Test plan is complete and approved by Ipreo and the client.
* Correct version is installed in performance testing environment, i.e. the version previously functionally tested and fixed if needed
* Test data is complete and in the performance testing environment in enough time to allow test scripts to be completed.
* Test accounts have been created in the performance testing environment in enough time to allow test scripts to be completed.
* Test scripts complete.
* All assigned resources are available to monitor the test.
* All parameter sets used in the script are generated based on the Database values.
  1. Exit Criteria
* All test scripts completed successfully
* No critical problems encountered
* All non-critical problems are logged
* All test logs are captured
  1. Suspension Criteria
* Not all test scripts will complete
* Critical problems are encountered and logged
* Hardware errors prevent the completion of the test

9. Responsibilities

Yaroslav Maslych – Performance Engineer

10. Schedule

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Stage** | **Period** | **Notes** |
| 1 | Test strategy design |  |  |
| 2 | Test plan creation |  |  |
| 3 | Draft of NFR definition |  |  |
| 4 | Setting up test environment |  |  |
| 5 | Test data preparation |  |  |
| 6 | Script/Scenarios design |  |  |
| 7 | Smoke, Capacity tests |  |  |
| 8 | Test results analysis |  |  |
| 9 | Test reporting |  |  |
| 10 | Load test and Stress test |  |  |
| 11 | Test results analysis |  |  |
| 12 | Test reporting |  |  |
| 14 | Scalability test and Stability test |  |  |
| 15 | Test results analysis |  |  |
| 16 | Test reporting |  |  |
| 17 | Volume test |  |  |
| 18 | Changing the data source to DB |  |  |
| 19 | Volume test |  |  |
| 20 | Test results analysis |  |  |
| 21 | Test reporting |  |  |
| 22 | Start regular tests |  |  |

11. Risk and contingencies

* Performance testing results can be essentially different even in case of minor difference in think times, arrival rate and test duration
* During the execution of the tests, some major performance or functional problems that may require code changes, creation of a new build may be discovered and in that case it may be necessary to repeat the load test from the beginning
* Performance testing tool is not capable of identically reproducing real life scenarios - so results could only be trusted as having limited reliability level
* Network/systems latency issues
* Environment’s temporary or constantly unavailability

11. Approves

Yaroslav Maslych - Approved