*“Optimizing the performance of computer systems has always been an art relegated to a few individuals who happen to have the ‘right skills’.”*

**Amir H. Majidimehr**  
Optimizing Unix for Performance, 1995



**Методика нагрузочного тестирования**

Версия 0.1

Оглавление

[1 История внесения изменений в документ 4](#_Toc127356998)

[2 Лист согласования 5](#_Toc127356999)

[3 Список терминов и сокращений 6](#_Toc127357000)

[4 Введение 7](#_Toc127357001)

[4.1 Назначение документа 7](#_Toc127357002)

[4.2 Объект тестирования 7](#_Toc127357003)

[5 Цели и задачи 9](#_Toc127357009)

[6 Ограничения тестирования 10](#_Toc127357010)

[7 Архитектура системы 11](#_Toc127357011)

[7.1 Конфигурация серверов продуктивного стенда 11](#_Toc127357012)

[8 Взаимодействие с внешними системами 12](#_Toc127357013)

[9 Стратегия тестирования 13](#_Toc127357014)

[9.1 Этапы тестирования 13](#_Toc127357015)

[10 Моделирование нагрузки 14](#_Toc127357016)

[11 Тестовый стенд 15](#_Toc127357017)

[11.1 Архитектура тестового стенда 15](#_Toc127357018)

[11.2 Конфигурация тестового стенда 15](#_Toc127357019)

[11.3 Конфигурация ПО 16](#_Toc127357020)

[11.4 Тестовые данные для средств НТ 16](#_Toc127357021)

[12 Анализ статистики 17](#_Toc127357024)

[*12.1* Наиболее часто используемые операции 17](#_Toc127357025)

[13 Профили нагрузки 19](#_Toc127357026)

[13.1 Процессинговая деятельность (MK) 19](#_Toc127357027)

[14 Сценарии пользования 20](#_Toc127357028)

[15 Наполнение БД 21](#_Toc127357029)

[16 Планируемые тесты 22](#_Toc127357030)

[16.1 Планируемые тесты 22](#_Toc127357032)

[16.2 Критерии успешности проведенного теста 22](#_Toc127357033)

[17 Требования к производительности 24](#_Toc127357034)

[18 Мониторинг производительности 25](#_Toc127357035)

[18.1 Метрики производительности 25](#_Toc127357036)

[18.2 Бизнес метрики 26](#_Toc127357037)

[18.3 Способы мониторинга показателей производительности 26](#_Toc127357037)

[19 Риски проекта 28](#_Toc127357038)

[20 Требования к заказчику 29](#_Toc127357039)

[21 Материалы, подлежащие сдаче 30](#_Toc127357040)

[22 Контакты 31](#_Toc127357046)

# История внесения изменений в документ

|  |  |  |  |
| --- | --- | --- | --- |
| Дата | Версия | Описание | Автор |
| 12.06.2024 | 0.1 | Документ создан | Бородулин Богдан  Зайнутдинов Марсель  Иванов Иван |

# Лист согласования

|  |  |  |  |
| --- | --- | --- | --- |
| ФИО | Должность | Подпись | Дата |
| Булатов Сергей | Менеджер НТ |  | 25.06.24 |

# Список терминов и сокращений

|  |  |
| --- | --- |
| Термин | Полное наименование |
| АРМ | Автоматизированное рабочее место |
| АС | Аппаратные средства |
| БД, DB | База данных |
| ИС | Информационная система |
| МП | Максимальная производительность |
| НТ | Нагрузочное тестирование |
| ОС | Операционная система |
| ПК | Программный комплекс |
| ПП | Программный продукт |
| СНТ | Средства нагрузочного тестирования |
| СУБД | Система управления базами данных |
| ТЗ | Техническое задание |
| ЧТЗ | Частное техническое задание |
| Boomq | Инструмент, предназначенный для автоматизации нагрузочного тестирования |
| Response Time, Rt | Время отклика |
| VUser, VU | Виртуальный пользователь |

Перечень общих терминов, наиболее часто использующихся в методиках, содержится в глоссарии:



# Введение

## Назначение документа

Методика НТ представляет собой подробное описание технологии нагрузочного тестирования систем программного комплекса «Boomq».

Документ преследует следующие цели:

* описать стратегию тестирования производительности систем;
* описать планируемые этапы работ;
* описать объект исследования и конфигурацию тестового стенда;
* описать порядок передачи результатов проекта;
* описать рамки и ограничения тестирования;
* описать методики тестирования.

Методика НТ предназначена для специалистов бизнес-подразделений, менеджеров и технических специалистов Заказчика, а также будет использоваться при проектировании и проведении тестов специалистами «Перформанс Лаб».

## Объект тестирования

Объектом тестирования является платформа Boomq Enterprise, предназначенная для автоматизации нагрузочного тестирования. Boomq состоит из следующих компонентов:

* **Frontend**. Пользовательский интерфейс Boomq;
* **Telegraf on Docker**. Агент для сбора метрик и данных о системе. Telegraf записывает собранные данные в TimescaleDB.
* **Proxy NGINX**. Прокси-сервер, который обеспечивает взаимодействие Frontend и Backend Boomq.
* **Микросервисы Boomq**. Микросервисы, которые обеспечивают функциональные возможности Boomq;
* **MinIO**. Файловое хранилище, в котором хранятся файлы, загружаемые пользователями, и файлы, которые создаются по результатам тестов, например, лог-файлы.
* **PostgreSQL**. База данных, которая содержит тестовые проекты, настройки и пользователей.
* **Grafana**. Инструмент для визуализации данных, полученных из TimescaleDB.
* **TimescaleDB**. База данных, которая содержит:
* системные метрики объекта тестирования;
* тестовые метрики, получаемые от JMeter;
* метрики нагрузочной станции, получаемые от Telegraf в контуре Boomq.

Экземпляров БД может быть несколько.

* **HAproxy**. Прокси-сервер, который обеспечивает взаимодействие платформы Boomq и TimescaleDB.
* **Patroni**. Приложение для управления репликацией между экземплярами TimescaleDB. Устанавливается на каждом экземпляре TimescaleDB.

Patroni регистрирует в Consul экземпляр TimescaleDB и получает информацию о других экземплярах БД. HAproxy получает актуальный список экземпляров TimescaleDB из Сonsul и данные о том, какие экземпляры поддерживает запись, а какие – чтение.

* **Consul**. Сервис, который регистрирует сервисы и экземпляры TimescaleDB, хранит настройки Boomq.
* **RabbitMQ**. Менеджер очередей для асинхронного взаимодействия микросервисов.
* **Нагрузочная станция с Docker daemon**. Подает нагрузку на объект тестирования. Может устанавливаться на сервер с микросервисами или на отдельный сервер. Нагрузочных станций может быть несколько. Станция включает в себя: **Docker c JMeter и Telegraf**. Docker-контейнер, который включает JMeter и Telegraf.

# Цели и задачи

Нагрузочное тестирование преследует следующие цели:

1. Выявление соответствия системы поставленным требованиям производительности.
2. Определение максимальной производительности системы и выявление узких мест компонент системы.
3. Тестирование системы на стабильность.

К основным задачам нагрузочного тестирования относятся:

* Анализ операций пользователей, проводимых в системе, с целью определения профилей нагрузки.
* Анализ базы данных для наполнения тестовыми данными.
* Наполнение базы данных.
* Разработка тестовой модели нагрузочного тестирования.
* Описание структуры стенда нагрузочного тестирования.
* Проведение испытаний в тестовой среде.
* Подготовка отчетов по результатам тестов.
* Поиск «узких мест» и подготовка рекомендаций по оптимизации производительности.

# Ограничения тестирования

В рамках проводимого нагрузочного тестирования следует отметить следующие ограничения:

* Данное тестирование не является функциональным и не служит для выявления функциональных дефектов, в то же время, обнаруженные в ходе проведения работ дефекты регистрируются и передаются Заказчику.
* Тестирование не направлено на выявление дефектов в аппаратной части стенда.
* Не оценивается влияние загруженности каналов связи.
* Перед проведением тестирования на этапе создания нагрузочных скриптов версии компонент информационной системы фиксируются и не изменяются до окончания тестирования, за исключением случаев устранения ошибок, мешающих дальнейшему проведению работ по тестированию.
* Специалисты Заказчика предоставляют профиль нагрузки.
* Организация, работоспособность и доступность тестового стенда обеспечивается Заказчиком.

# Архитектура системы

Boomq Enterprise – автономное приложение аналогичное Boomq Cloud, которое устанавливается и работает на серверах клиента. На рис. 1 представлена архитектура данного приложения.Компоненты Boomq Enterprise описаны в главе [4.2 Объект тестирования](#_Объект_тестирования)

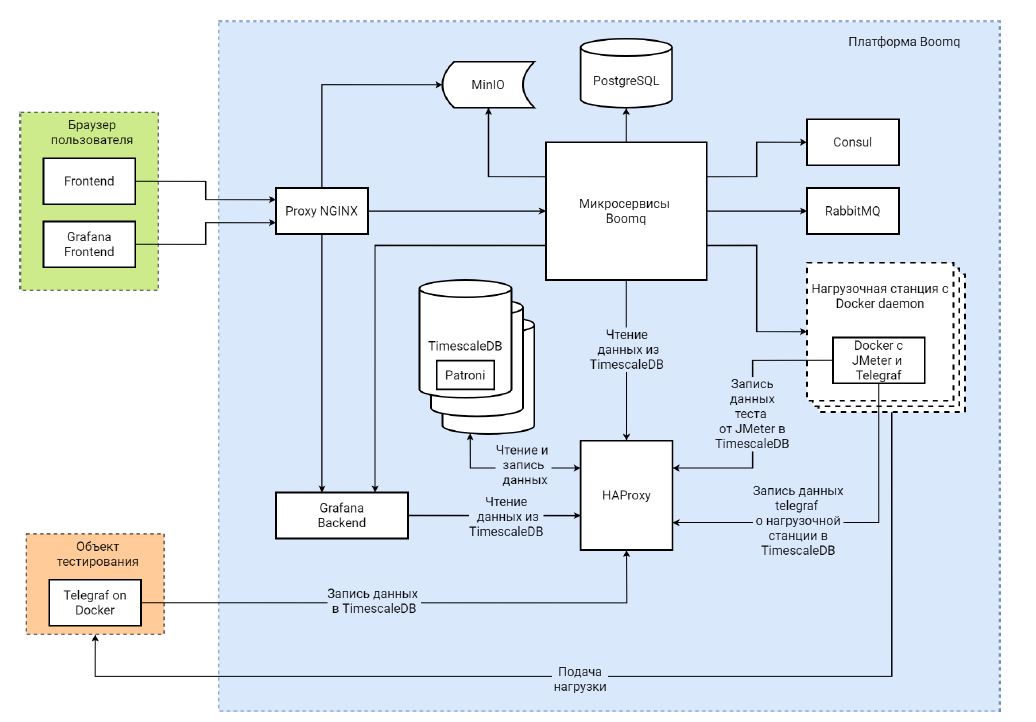


Рисунок 1 – Компонентная архитектура Boomq

## Конфигурация серверов продуктивного стенда

Раздел содержит таблицу с аппаратными характеристиками (марка сервера, тип и количество CPU, память, дисковый массив, операционная система, используемая СУБД, сервер приложений и т.д.).

Таблица 1 – Конфигурация серверов продуктивного стенда

| **СИСТЕМА** | **ПАРАМЕТР** | **ЗНАЧЕНИЕ** |
| --- | --- | --- |
| Boomq | CPU cores | 8 |
| RAM | 16 GB |
| Hard | 100GB |
| Software | OS Centros 7, Boomq Enterprise 3.2.1 |
| Нагрузочная станция Boomq | CPU cores | 8 |
| RAM | 32 GB |
| Hard | 500GB |
| Software | OS Centros 7 |

# Взаимодействие с внешними системами

**Внешняя нагрузочная станция 77.50.236.215:22022.** Для сравнения результатов тестирования, кроме внутренней нагрузочной станции платформы Boomq (Docker Daemon) будет использоваться внешняя нагрузочная станция 77.50.236.215 (порт 22022). Нагрузочные станции осуществляют отправку запросов в специальный эмулятор, именуемый далее, как “Заглушка”.

**Заглушка.** Система Boomq будет осуществлять тесты отправляя запросы на специально разработанную заглушку, которая в свою очередь отправляет сообщения в распределенную систему Kafka.

**Kafka.** Принимает и обрабатывает сообщения от заглушки.

Таблица 2 – Перечень сервисов для эмуляции нагрузки

|  |  |  |
| --- | --- | --- |
| сервис | Функциональность | Внешние системы |
| Внешняя нагрузочная станция | Альтернативный источник нагрузки | 77.50.236.215:22022 |
| Заглушка | Получение запросов из Boomq | 77.50.236.215:20022 |

# Стратегия тестирования

## Этапы тестирования

Проект проведения нагрузочного тестирования делится на следующие этапы:

* создание методики (данный документ);
* разработка средств нагрузочного тестирования (СНТ);
* разработка генерации данных для тестирования на увеличенных объемах.
* проведение тестирования;
* системный анализ;
* подготовка отчета.

Критерии успешного завершения тестирования:

* достигнута повторяемость результатов тестов;
* выполнены все запланированные тесты;
* получены данные мониторинга;

# Моделирование нагрузки

Средства НТ разрабатываются с использованием ПО LoadRunner (version 2021 Build 371), предназначенного для создания тестов и проведения тестирования.

Эмулятор системы принимающей сообщения от Boomq (“Заглушка”) разрабатывался на JDK 17 Coretto.

Моделирование нагрузки от операционной деятельности производится с использованием средств НТ, путем эмуляции действий пользователя на платформе Boomq по протоколу HTTPS c помощью LoadRunner. В свою очередь Boomq отправляет запросы в ”Заглушку” по протоколу HTTPS с помощью внутренней нагрузочной станции Docker Daemon, либо с помощью внешней нагрузочной станции 77.50.236.215 “Заглушка” отправляет сообщения в Kafka.

Величина задержки и количество виртуальных пользователей, выполняющих различные сценарии, рассчитываются с использованием Excel шаблона на этапе подготовки стенда и средств НТ после написания скриптов и определения времени их работы в ИС, не испытывающей нагрузку.

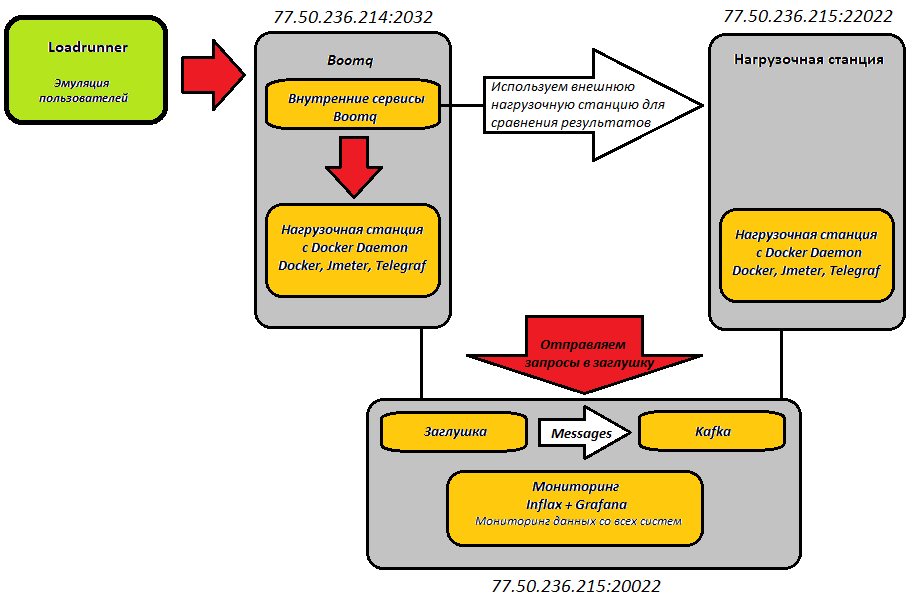


Рисунок 2 - Схема генерации нагрузки

# Тестовый стенд

## Архитектура тестового стенда

Архитектура тестового стенда (рис. 3) включает в себя платформу Boomq с внутренней нагрузочной станцией (77.50.236.214:2032), дополнительную внешнюю нагрузочную станцию (77.50.236.215:22022) для получения возможности сравнения результатов, сервер с эмулятором обработчика (77.50.236.215:20022), брокером сообщений Kafka, а также мониторингом системных метрик.

Запись метрик всех систем участвующих в тестировании с помощью сервисов Telegraf, InfluxDB, Grafana. Telegraf установлен на каждой системе, снимает системные метрики и отправляет данные в InfluxDB, который в свою очередь хранит и организует полученные данные с временными отметками в виде серий. Grafana позволяет создать настраиваемые графики на основе временных рядов данных хранящихся в InfluxDB.

Взаимодействие с системой Boomq будет реализовано в виде эмулятора файлового обмена, подробнее можно посмотреть в разделе Моделирование нагрузки.



Рисунок 3 – Архитектурная схема тестового стенда

## Конфигурация тестового стенда

В таблице 3 указана конфигурация тестового стенда.

Таблица 3 – Конфигурация тестового стенда

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Тип/имя сервера | Конфигурация | Кол-во в прод. | Кол-во для нагрузочного тестирования | ОС |
| Тестовый стенд  IP: 77.50.236.214:2032 | 8x Intel® Xeon® CPU E5-2690v2 @ 3.00 GHz, 16 GB, HDD 100Gb | - | 1 | Centos 7, |
| Сервер заглушек  IP: 77.50.236.215:20022 | 8x Intel® Xeon® CPU Gold 6326 CPU @ 2.90GHz, 32GB, 200Gb | - | 1 | Ubuntu 22.04.3 LTS |
| Сервер с нагрузочными станциями Boomq  IP: 77.50.236.215:22022 | 8x CPU, 32GB, 500 Gb | - | 1 | Centos 7 |

## Конфигурация ПО

Тестирование проводится на системе Boomq Enterprise 3.2.1, ОС Centos 7 .

Сервер с нагрузочными станциями Boomq , ОС Centos 7.

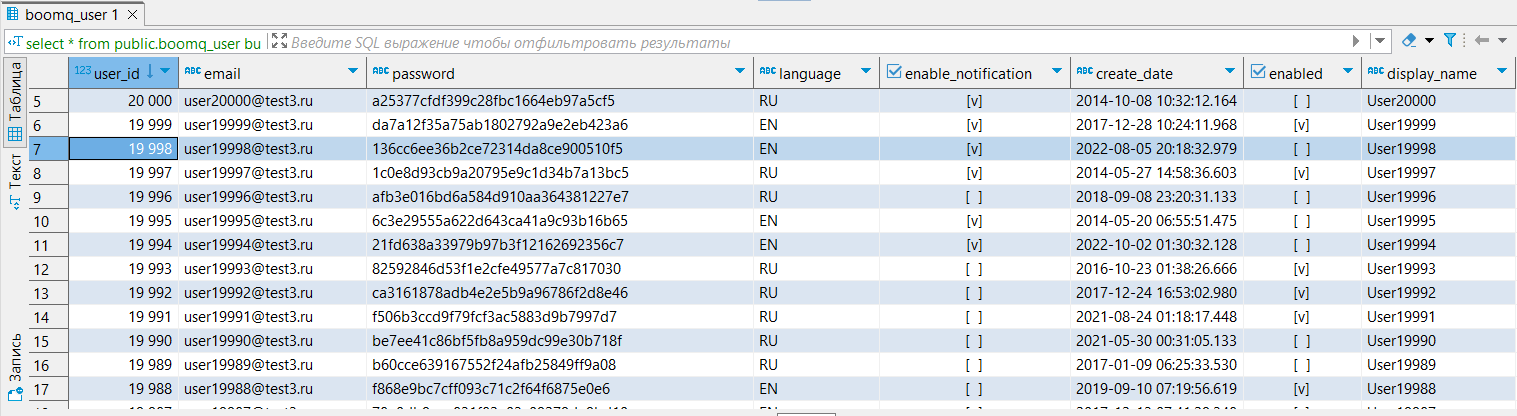
Сервер заглушек, ОС Ubuntu 22.04.3 LTS.

## Тестовые данные для средств НТ

Для выполнения требований заказчика по наполнения БД, с помощью нагрузочных скриптов были сгенерированы данные новых пользователей. Данные включают в себя следующие поля:

* *user\_id;*
* *email;*
* *password;*
* *language*
* *enable\_notification;*
* *create\_data;*
* *enabled;*
* *display\_name.*

Пример сгенерированной записи:



# Анализ статистики

## Наиболее часто используемые операции

На основании проведенного анализа бизнес процессов и анализа предоставленной Заказчиком статистики выполнения операций в промышленной системе за два месяца с 01.01.2023 по 27.02.2023 был выбран день с максимальной интенсивностью и час пиковой нагрузки (ЧПН) с 17:00 до 18:00 и определены наиболее часто используемые и ресурсоемкие операции.

Рисунок 4 – График распределения операций по времени за сутки

В таблице ниже представлен список наиболее часто используемых операций.

Таблица 4 – Список операций с разбивкой по типам по типам в ЧПН

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Названия операций** | **Количество** | **% от общего количества** | **% с нарастающим итогом** |
| 1 | Запустил Test | 85 | 26.90% | 26.90% |
| 2 | Стартовал Grafana platform | 80 | 25.32% | 52.22% |
| 3 | Перешел в status CANCELED | 70 | 22.15% | 74.37% |
| 4 | Перешел в status TEST\_STOPPING | 70 | 22.15% | 96.52% |
| 5 | Перешел в status FINISHED | 11 | 3.48% | 100% |
| 6 | Перешел в status FAILED | 0 | 0% | 100% |
| 7 | Подписка для команды создана | 0 | 0% | 100% |
| 8 | Команда создана в Графане | 0 | 0% | 100% |
| 9 | Команда создана | 0 | 0% | 100% |
| 10 | Зарегистрировался | 0 | 0% | 100% |
| 11 | Ошибка регистрации | 0 | 0% | 100% |
| 12 | Обновил подписку | 0 | 0% | 100% |
|  | **Общий итог** | **316** | **100%** |  |

Зеленым отмечены операции, выбранные для профиля нагрузки «Операционная деятельность».

# Профили нагрузки

## Пиковый день

Профиль “Пиковый день” включает себя операции запуска, отмены или остановки пользовательских тестов на платформе Boomq, а также старт отрисовки метрик в Grafana.

Профиль сформирован на основе анализа статистики использования системы Boomq, основную нагрузку на систему обеспечивают 4 типа операции указанных в таблице 6.

Таблица 6

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Название операции | Распределение % | Средняя нагрузка | Пиковая нагрузка в час |
| 1 | Запустил Test | 26,30 % | 30,6 | 85 |
| 2 | Стартовал Grafana platform | 25,34 % | 29,5 | 80 |
| 3 | Перешел в status CANCELED | 22,98 % | 26,8 | 70 |
| 4 | Перешел в status TEST\_STOPPING | 22,98 % | 26,8 | 70 |

# Сценарии пользования

Для эмуляции процесса были записаны скрипты, соответствующие выполнению следующих пользовательских операций:

Таблица 6 – Перечень эмулируемых операций

|  |  |  |
| --- | --- | --- |
| ID теста | Название операции | Название скрипта |
| UC01 | Регистрация пользователя | UC01\_User\_registration |
| UC02 | Создание теста | UC02\_Create\_test |
| UC03 | Запуск теста | UC03\_Start\_test |
| UC04 | Отмена теста | UC04\_Stop\_test |
| UC05 | Создание тренда | UC05\_Create\_trend |
| UC06 | Создание отчета | UC06\_Create\_report |

Примечание. При эмуляции, каждая операция будет в свою очередь разбита на транзакции (логин, открытие списка тестов, и др.).

# Наполнение БД

В таблице boomq\_user БД postgres изначальное количество записей составляло 1890. По требованию Заказчика была выполнена генерация данных до 20000 записей в таблице boomq\_user БД postgres. Для решения данной задачи был разработан генератор записей. Генерация новых данных выполнена с помощью запроса INSERT INTO PostgreSQL:

*INSERT INTO boomq\_user (user\_id, email, "password", "language", enable\_notification, create\_date, enabled, display\_name)*

*SELECT*

*1894 + generate\_series, -- Генерация уникального user\_id*

*'user' || (1894 + generate\_series) || '@test3.ru', -- Генерация email c доменом @test3.ru*

*md5(random()::text), -- Генерация уникального password*

*CASE floor(random() \* 2) -- Генерация языка: RU или EN*

*WHEN 0 THEN 'EN'*

*ELSE 'RU'*

*END,*

*CASE floor(random() \* 2) -- Генерация булевого значения колонки enable\_notification*

*WHEN 0 THEN true*

*ELSE false*

*END,*

*to\_char(now() - (random() \* 10 \* interval '1 year') - (random() \* 365 \* interval '1 day') - (random() \* 24 \* interval '1 hour'), 'YYYY-MM-DD HH24:MI:SS.MS'), -- Генерация даты за последние 10 лет*

*CASE floor(random() \* 2)*

*WHEN 0 THEN true*

*ELSE false*

*END, -- Генерация булевого значения колонки enabled*

*'User' || (1894 + generate\_series) -- Генерация уникального display\_name*

*FROM generate\_series(1, 20000 - 1890); -- 20000 - 1890 = 18110 итого будет дополнительных строк*

# Планируемые тесты

## Планируемые тесты

|  |  |
| --- | --- |
| **Название** | **Описание** |
| **Определение максимальной производительности** | При тестировании происходит пошаговое увеличение нагрузки с нуля до предельной (с шагом 20% от плановой). Пошаговое увеличение происходит до тех пор, пока не нарушится один из критериев успешности. Время работы теста на каждом шаге (ступени) после стабилизации нагрузки (этап стабилизации нагрузки равен 5 минутам) составляет 10 мин. По результатам устанавливается уровень нагрузки L0 (последняя ступень нагрузки, на которой не были нарушены критерии успешности) |
| **Подтверждение максимальной производительности** | Тест проводится на ступени нагрузки, предшествующей L0 (или на уровне нагрузки 90% от L0). Длительность стабильной нагрузки не менее 1 часа. Если в процессе тестирования система оказалась недогружена или перегружена, то значение нагрузки корректируется и второй тест проводится повторно. В случае увеличения нагрузки новый уровень может быть рассчитан на основе данных об утилизации ресурсов. Результатом тестирования является максимальный достигнутый уровень нагрузки (обозначается Lmax) |
| **Тест стабильности системы** | Тест проводится на уровне нагрузки Lstab = 80% от Lmax. Длительность стабильной нагрузки не менее 4 часов. В ходе теста фиксируются все отклонения от "нормального" поведения системы, в т.ч. деградация производительности, утечки памяти |

## Критерии успешности проведенного теста

Тест считается успешным, если:

* в процессе тестирования запросы выполнялись с частотой, соответствующей профилю тестирования (с возможным отклонением не более 10%);
* в процессе тестирования возникло не более 5% ошибок
* по окончании теста получены данные по временам отклика системы и по использованию системных ресурсов и соответствуют требованиям производительности;
* критерии проверяются по данным, полученным за интервал стабилизированной нагрузки в соответствии с длительностью теста.

# Требования к производительности

Таблица 11 – Требования к времени отклика по типам операций

|  |  |
| --- | --- |
| ОПЕРАЦИЯ | ТРЕБОВАНИЕ К ВРЕМЕНИ ОТКЛИКА (СЕК) |
| Регистрация пользователя | 3 |
| Создание теста | 3 |
| Запуск теста | 3 |
| Отмена теста | 3 |
| Создание тренда | 3 |
| Создание отчета | 3 |

Также в качестве требований к производительности системы определяются:

* средняя утилизация процессорных мощностей системы не должна превышать 80% (рекомендуемый максимум);
* средняя утилизация дисковой подсистемы должна не превышать 80%
* утилизация RAM оперативной памяти не должна превышать 80%
* система должна быть способна обработать операции с интенсивностью, указанной в профиле нагрузки (см. раздел Моделирование нагрузки);
* отсутствие ошибок, связанных с недоступностью системы, при выполнении операций.

# Мониторинг производительности

Мониторинг производительности серверов под управлением ОС Centos 7, ОС Ubuntu 22.04.3 LTS, ОС Centos 7 будет включать наблюдение за системными параметрами и будет осуществляться при помощи таких утилит Telegraf, InfluxDB, Grafana. Интервал сбора статистики по метрикам мониторинга будет составлять 15 секунд.

Параметры производительности системы на стороне нагрузочных станций будут измеряться при помощи встроенных средств Load Runner

## Метрики производительности

Для Unix-систем, используются следующие метрики:

Общая загрузка CPU:

* Time — время (ДД.ММ.ГГГГ ЧЧ24:МИ:00);
* %usr — активность прикладных программ;
* %sys — обслуживание системных вызовов;
* %wio — время, затраченное на ожидание ввода/вывода;
* %load — загрузка CPU. Вычисляется как (100 – %idle).

Загрузка CPU с разбиением по ядрам:

* Time — время (ДД.ММ.ГГГГ ЧЧ24:МИ:00);
* #cpu — ID ядра;
* %usr — активность прикладных программ;
* %sys — обслуживание системных вызовов;
* %wio — время, затраченное на ожидание ввода/вывода;
* %load — загрузка CPU. Вычисляется как (100 – %idle).

Длина очереди CPU:

* Time — время (ДД.ММ.ГГГГ ЧЧ24:МИ:00);
* runq-sz — длина очереди.

Количество контекстных переключений:

* Time — время (ДД.ММ.ГГГГ ЧЧ24:МИ:00);
* cswitch/s — количество переключений.

Базовые метрики дисков:

* Time — время (ДД.ММ.ГГГГ ЧЧ24:МИ:00);
* device — ID диска (LUN);
* %busy — % времени работы диска (загрузка);
* avque — дисковая очередь;
* avwait — время ожидания в очереди, мс;
* avserv — время непосредственной работы диска, мс;
* iotime — avwait + avserv.

Метрики пропускной способности дисков:

* Time — время (ДД.ММ.ГГГГ ЧЧ24:МИ:00);
* device — ID диска (LUN);
* iops — количество операций чтения/записи;
* kb/s — объем считанных и записанных данных, KiB.

Файл подкачки:

* Time — время (ДД.ММ.ГГГГ ЧЧ24:МИ:00);
* swap\_in — байт закачано;
* swap\_out — байт выкачано.

Утилизация ОП:

* Time — время (ДД.ММ.ГГГГ ЧЧ24:МИ:00);
* UsedMem — используемая ОП, MiB;
* FreeMem — свободная ОП, MiB;
* UsedMem% — % утилизации ОП.

Все показатели снимаются и анализируются за репрезентативные периоды выполнения тестов:

* за периоды плановой нагрузки на систему (без учета периодов теста, на которых происходит выход на нагрузку);
* за периоды насыщения системы и деградации производительности, если таковая будет иметь место.

## Бизнес метрики

При тестировании используются следующие бизнес-метрики:

* Количество ошибок;
* Интенсивность;
* Время отклика.

## Способы мониторинга показателей производительности

Для снятия показателей производительности будут использоваться следующие инструменты мониторинга:

Таблица 12 – Типы метрик производительности

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Тип метрики** | **Способ снятия** | **Способ получения** |
| 1 | Аппаратные ресурсы | Предустановленные в ОС средства мониторинга утилизации аппаратных ресурсов: Telegraf, Infuxdb, Grafana | Предоставляются специалистами заказчика по запросу исполнителя.  Инструкции и скрипты для сбора статистики будут предоставлены исполнителем при подготовке к проведению тестирования. |
| 2 | Бизнес-показатели | Встроенные в инструмент тестирования счетчики, данные логов( LoadRunner). Данные отправляются в Infuxdb | Фиксируются исполнителем. |

# Риски проекта

В разделе указываются все возможные риски, которые могут повлиять на скоуп, трудозатраты, сроки, достижимость и корректность результата.

* Методика тестирования может быть пересмотрена и скорректирована (по результатам получения актуальной информации, актуальных целей и т.д.). Все изменения предварительно согласовываются с заказчиком.
* В случае невозможности генерации требуемых объемов БД в поставленные сроки, по согласованию с заказчиком, тестирование будет проведено на меньших объемах (которые могут быть сгенерированы в требуемые сроки).

# Требования к заказчику

В разделе описываются требования:

* Три рабочих места с доступами к нагрузочным станциям.
* Нагрузочные станции для разработки скриптов и подачи нагрузки (подробнее см. в разделе Моделирование нагрузки).
* Консультации аналитиков или подробные, исчерпывающие инструкции по выполнению пользовательских операций.
* Статистику использования системы в часы пиковой нагрузки с группировкой по времени в разрезе типов операций (количество операций каждого типа, например, в час). Или доступ к источникам данных, из которых возможно сделать данную статистическую выборку.
* Прогнозы роста нагрузки на систему, если таковые имеются.
* Прогнозы роста объемов ключевых сущностей БД (пользователи) если они имеются.
* Спецификация тестового стенда (аппаратная конфигурация серверов системы, архитектурная схема расположения компонент системы на серверах, версии устанавливаемого ПО).
* Спецификация аппаратно-программного комплекса, планируемого для внедрения в промышленную эксплуатацию (аппаратная конфигурация серверов системы, архитектурная схема расположения компонент системы на серверах, версии устанавливаемого ПО).
* Заказчик организовывает исправление блокирующих дефектов, обнаруженных Исполнителем, таким образом, чтобы это не приводило к простаиванию (частичному или полному) представителей Исполнителя, занятых выполнением Работ.
* Заказчик предоставляет Исполнителю права доступа к системе на уровне, достаточном для проведения тестирования системы и снятия необходимых метрик производительности.

# Материалы, подлежащие сдаче

В разделе приводятся материалы, которые необходимо предоставить заказчику в процессе ведения проекта. Данные предоставляются в виде таблицы с указанием названия материала, формата предоставления, при необходимости также может указываться срок сдачи материалов. Не зависимо от целей и задач тестирования обязательными к предоставлению являются исходные коды скриптов нагрузочного тестирования и результаты всех проведенных тестов.

|  |  |  |
| --- | --- | --- |
| **№** | **Наименование** | **ФОРМАТ ПРЕДОСТАВЛЕНИЯ** |
| 1 | Исходные коды скриптов нагрузочного тестирования | zip (архив) , ссылка GitHub |
| 2 | Парсер для извлечения и анализа предоставленных заказчиком логов. | txt, ,ссылка GitHub |
| 3 | Эмулятор (заглушка) | MS Word, ссылка GitHub |
| 4 | Отчет по результатам тестирования (включая выводы и рекомендации) | MS Word, ссылка GitHub |
| 5 | Методика нагрузочного тестирования | MS Word, ссылка GitHub |
| 6 | Результаты всех проведенных тестов | MS Word, ссылка GitHub |

# Контакты

ООО «Перфоманс Лаб»

121087 Москва, ул. Барклая, 6, стр.5, офис 511

Телефон: +7 495 780 9228

Факс: +7 495 780 9228

[http://performance-lab.ru](http://performance-lab.ru/)

Генеральный директор: Кутузов Максим Юрьевич

Документ подготовили:

Бородулин Богдан Владимирович Студент

Зайнутдинов Марсель Миниахметович Студент

Иванов Иван Сергеевич Студент