*“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



**Финальный отчёт**

Оглавление

[1 История внесения изменений в документ 3](#_Toc160298378)

[2 Лист согласования 4](#_Toc160298379)

[3 Список терминов и сокращений 5](#_Toc160298380)

[4 Назначение документа 6](#_Toc160298381)

[5 Основные положения 7](#_Toc160298382)

[5.1 Объект тестирования 7](#_Toc160298383)

[5.2 Цели и задачи 8](#_Toc160298384)

[6 Выводы 9](#_Toc160298385)

[7 Особенности тестируемой системы 10](#_Toc160298386)

[8 Обнаруженные проблемы 11](#_Toc160298387)

[9 Рекомендации 13](#_Toc160298388)

[10 Методика тестирования 15](#_Toc160298389)

[10.1 Профили нагрузки 15](#_Toc160298390)

[10.2 Эмуляция нагрузки 16](#_Toc160298391)

[10.3 Критерии успешности проведения тестов 16](#_Toc160298392)

[10.4 Проведённые тесты 16](#_Toc160298393)

[10.5 Ограничения тестирования 19](#_Toc160298394)

[11 Результаты нагрузочного тестирования 20](#_Toc160298395)

[11.1 Результаты проведения теста поиска максимальной производительности 20](#_Toc160298396)

[11.2 Результаты проведения теста подтверждения максимальной производительности 23](#_Toc160298397)

[11.3 Результаты проведения теста с увеличением нагрузки до 150% от найденного максимума 25](#_Toc160298398)

[11.4 Результаты проведения теста при отказе БД 29](#_Toc160298399)

[11.5 Результаты проведения теста сбоя приложения 32](#_Toc160298400)

[11.6 Результаты проведения теста определения надежности 35](#_Toc160298401)

[12 Приложение 39](#_Toc160298402)

[13 Контакты 41](#_Toc160298403)

# История внесения изменений в документ

|  |  |  |  |
| --- | --- | --- | --- |
| Дата | Версия | Описание | Автор |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Лист согласования

|  |  |  |  |
| --- | --- | --- | --- |
| ФИО | Должность | Подпись | Дата |
| Матвеев Александр Павлович |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Список терминов и сокращений

|  |  |
| --- | --- |
| Термин | Полное наименование |
| НТ | Нагрузочное тестирование |
| БД, DB | База данных |
|  |  |
|  |  |
|  |  |

# Назначение документа

Данный документ представляет собой отчет по результатам нагрузочного тестирования систем сервиса «UXCrowd», проведенного в рамках обучающего проекта «Перфоманс Лаб» в период с 02.2024 по 03.2024 г. Документ содержит краткую методику, результаты нагрузочного тестирования, выводы о производительности системы и рекомендации по оптимизации.

# Основные положения

## Объект тестирования

Объектом тестирования является облачный сервис «UXCrowd» для удаленных немодерируемых тестирований на пользователях.

Основная нагрузка на систему создается путем эмуляции операционной деятельности (пользовательские операции):

1. **Регистрации тестировщика**
2. **Регистрация клиента**
3. **Создание новых тестов**
4. **Прохождение новых тестов**

Подробнее объект тестирования и тестовая модель описаны в методике нагрузочного тестирования.

## Цели и задачи

Нагрузочное тестирование преследует следующие цели:

1. Определение максимальной нагрузки, которую система может выдержать без значительного снижения производительности.
2. Оценка надежности и устойчивости системы при ограниченной или повышенной нагрузке.
3. Тестирование системы на стабильность.
4. Выявление узких мест, проблем производительности, сбоев или неожиданного поведения системы под нагрузкой.
5. Анализ существующих проблем тестируемой среды. Предоставление рекомендаций по оптимизации.

# Выводы

На основе результатов тестирования был произведён анализ, который позволяет сделать следующие выводы:

1. Определение максимальной нагрузки.

* Проведение тестов показало, что изменение количества одновременных пользователей значительно влияет на стабильно системы.
* Результаты тестирования показали, что системные требования тестового стенда не удовлетворяют требованиям производительности для 100% профиля нагрузки.
* Максимальная производительность достигнута при нагрузке 6% от текущего профиля нагрузки. Шаг увеличения подачи нагрузки составляет 2%. При нагрузке 8% от текущего профиля нагрузки наблюдается резкий скачок по загрузки CPU. Наибольшую нагрузку на CPU создает сценарий создания теста (UC\_03\_new\_test\_creating) по причине не оптимизированного запроса к БД. Более подробная информация представлена в разделе 11.1.
* Во время проведения теста подтверждения максимума при 6% от текущего профиля нагрузки нагрузка CPU составила 72,18% , что не соответствует критериям успешности.
* Для проведения теста подтверждения максимума принята нагрузка 4% от текущего профиля нагрузки. При выбранном уровне нагрузки критерии надежности нарушены не были. Более подробная информация представлена в разделе 11.2.

1. Оценка надежности и устойчивости системы.

* При повышении нагрузки до 150% от найденного максиму и при отключении (сбое) приложения, система вернулась в нормальное состояние. Более подробная информация представлена в разделах 11.3 и 11.5.
* После отключения БД и возобновлении ее работоспособности, наблюдается значительный рост времени отклика некоторых запросов, которые со временем не возвращаются к нормальным значениям. Более подробная информация представлена в разделе 11.4.

1. Тестирование системы на стабильность.

Во время проведения теста стабильности при 70% от максимальной нагрузки на протяжении 11 часов наблюдалась тенденция увеличения времени откликов запросов, Увеличение времени работы приложения при нагрузке приведет к нарушению критериев успешности. Более подробная информация представлена в разделе 11.6.

1. Выявление узких мест, проблем производительности.

В процессе тестирования были выявлен ряд проблем, основными из которых можно отметить утечка памяти и загрузка CPU при работе Garbage Collector. Подробное описание об этих и других проблемах представлено в разделе 8.

1. Предоставление рекомендаций по оптимизации приведены в разделе 9

# Особенности тестируемой системы

Подача нагрузки на систему тестового стенда отличается от подачи нагрузки на продуктовый стенд. В процессе тестирования были выявлены следующие особенности:

1. **Описание:** отсутствие реализации сервиса для отправки email-сообщений при регистрации тестировщика и клиента. Пароль для входа зарегистрированного пользователя на почту не направляется;

**Статус:** решена;

**Способ решения:** для выполнения входа в учетную запись с помощью JDBC Request была реализована смена пароля напрямую в БД  у созданных пользователей.

1. **Описание:** проблема с прохождением тестов у новых тестировщиков. При регистрации нового тестировщика, ему присваивается роль “ROLE\_NEW\_TESTER”, что накладывает ограничения на прохождение тестов.

**Статус:** решена;

**Способ решения:** реализована замена роли тестировщика в БД с “ROLE\_NEW\_TESTER” на “ROLE\_TESTER” в нагрузочных сценариях.

1. **Описание:** проблема с конвертацией видео в четвертом (UC\_04\_new\_test\_pass) сценарии.

**Статус:** решена частично;

**Способ решения:**

* для прохождения четвертого сценария (UC\_04\_new\_test\_pass) в качестве загружаемого видео для тестирования использовался видеофайл с минимальным для требования системы размером 55 КБ.
* выполнена проверка успешной конвертации файлов в БД.
* для временного решения проблемы утечки памяти был выполнен перезапуск контейнеров с приложением.

1. **Описание:** проблема с генерацией интервью-тестов без шагов выполнения, необходимых для корректного прохождения четвертого сценария.

**Статус:** решена;

**Способ решения:** перед прохождением сценариев в БД было проведено изменение статусов на “COMPLETED” для интервью-тестов, не имеющих шагов выполнения.

# Обнаруженные проблемы

В процессе тестирования были выявлены следующие проблемы:

1. **Описание проблемы:** проблема долгой загрузки всех исследований у клиентов. Загрузка страницы, отображающая все доступные исследования, производится порядка 40 секунд. Причиной является недостаток объема памяти, выделяемый для операций сортировки и хэширования данных.

**Статус:** решена;

**Способ решения проблемы:**

* увеличен объем памяти, выделяемый для операций сортировки и хэширования данных в рамках одного рабочего процесса work\_mem c 4 MB до 64 MB, что привело к ускорению выполнения запроса;
* увеличен объем выделенной памяти для хранения общих данных shared\_buffers со 128 MB до 256 MB, что позволило БД использовать больше памяти для более эффективной работы с общими данными (дальнейшее увеличение памяти не повлияло на быстродействие системы);
* увеличен размер буфера журнала транзакций wal\_buffers с 4 MB до 64 MB, что позволило улучшить производительность базы данных при записи и подтверждении транзакций (дальнейшее увеличение памяти не повлияло на быстродействие системы);
* уменьшен объем максимальной памяти для кэширования данных доступной операционной системе effective\_cache\_size с 4 GB до 64 MB, что привело к уменьшению оценки доступного кэша для хранения данных;

1. **Описание проблемы:** проблема утечки памяти, связанная с работой java приложения. что приводит к аномальным скачкам загрузки CPU из-за активности Garbage Collector, который при работе использует все ядра процессора. На этапе логирования уведомления пользователя об оплаченном тесте возникают блокировки доступа к общему ресурсу, что влияет на работу сборщика мусора. Сборщик мусора работает в фоновом режиме, периодически освобождая память от объектов, на которые больше не ссылается программа. Если доступ к общему ресурсу заблокирован на длительное время из-за логирования, это приводит к накоплению неуправляемых объектов в памяти и увеличению потребления ресурсов.

**Статус:** не решена;

**Описание проблемы:** Проблема при создании учетной записи клиента в БД для нее загружается шаблонный json-файл с интервью, даже если интервью для данного пользователя не создавалось. Вес шаблонного json-файл составляет 60 Кб. Рекомендации по решению данной проблемы приведены в п. 9.

**Статус:** не решена;

1. **Описание проблемы:** проблема с конвертацией видео в четвертом (UC\_04\_new\_test\_pass) сценарии. При загрузке видео размером более 1 МБ, большую часть нагрузки на процессор в четвертом сценарии занимает утилита ffmpeg. Данная утилита вызывается внутри контейнера java-приложением. При загрузке видео файл конвертируется из формата webm в формат mp4. После конвертации файлов наблюдается дублирующее хранение исходного и конвертированного файла. Рекомендации по решению данной проблемы приведены в п. 9.

**Статус:** не решена.

# Рекомендации

1. Провести ручную оптимизацию автоматически сгенерированных SQL-запросов. (один  из проблемных запросов приведен в приложении А.)

Создать индексы на колонках, которые используются в условиях сравнения (WHERE) и объединения таблиц (JOIN). Например, индексы на колонках, по которым происходят соединения: id в таблицах customer\_orders, participant\_group, orders, tester\_task, users, customer, customer\_orders, part\_group\_social\_statuses, part\_group\_educations. Пересмотреть подзапросы и попробовать упростить их логику или заменить на JOIN, если это возможно. Например, объединить таблицы вместо использования подзапросов для подсчета значений.

1. Провести корректировку параметров конфигураций БД.

увеличить объем памяти, выделяемый для операций сортировки и хэширования данных в рамках одного рабочего процесса work\_mem, что приведет к ускорению выполнения запроса;

увеличить объем выделенной памяти для хранения общих данных shared\_buffers, что позволит БД использовать больше памяти для более эффективной работы с общими данными;

Увеличить размер буфера журнала транзакций wal\_buffers, что позволит улучшить производительность базы данных при записи и подтверждении транзакций ;

Уменьшить объем максимальной памяти для кэширования данных доступной операционной системе effective\_cache\_size, что приведет к уменьшению оценки доступного кэша для хранения данных.

1. Провести оптимизацию загрузки тяжеловесных шаблонных json-файлов, которые создаются  при создании учетной записи клиента.

Оптимизировать создание шаблонного json-файла, исключив лишнюю информацию.

Реализовать подгрузку json-файлов только при необходимости, например, когда пользователь запросит подробности конкретного исследования.

Выполнить разбиение больших json-файлов на более мелкие.

Реализовать сжатие json-файлов с помощью алгоритмов сжатия данных (например, gzip) перед передачей их по сети. Это поможет уменьшить объем передаваемых данных и ускорить загрузку.

1. Оптимизировать работу java-приложения путем настройки работы механизма Garbage Collector.

Использовать Xms и Xmx параметры для определения начального и максимального размера heap. Это поможет избежать фрагментации памяти и снизить частоту срабатывания Garbage Collector.

Настроить параметр  -XX:GCTimeRatio,  позволяющий управлять балансом между временем, затрачиваемым на сборку мусора, и доступной памятью.

Используйте мониторинг и профилирование: следите за работой сборщика мусора, чтобы своевременно обнаруживать проблемы с производительностью и оптимизировать их.

1. В личном кабинете тестировщика для уменьшения времени загрузки списка исследований реализовать пагинацию.
2. Исключить из работы приложения создание заказов со статусом ‘DRAFT’ при регистрации клиента. Или хранить данные заказы в отдельной от остальных заказов таблице БД
3. Выделить отдельный сервер для системы мониторинга.  Установить на новом сервере инструменты мониторинга, настроить соединение между системой мониторинга и основным приложением, чтобы система могла точно отслеживать его состояние.
4. Провести корректировку работы утилиты ffmpeg, которая отвечает за загрузки и конвертацию видео.

Ограничить доступ к ресурсам для утилиты ffmpeg. Внедрите мониторинг для отслеживания ресурсов, используемых утилитой ffmpeg. Это поможет идентифицировать проблемные места и возможные точки оптимизации.

Поскольку конвертация видео не является первоочередной бизнес задачей, производить работу конвертации видео в моменты, когда нагрузка CPU минимальна (например в ночное время). И производить конвертацию видео по запросу, если видео еще не было конвертировано.

После конвертации видео, рассмотреть возможность удаления исходного файла формата webm. Это может сэкономить дисковое пространство и уменьшить нагрузку на систему.

1. На этапе логирования уведомления пользователя об оплаченном тесте использовать асинхронное логирование: вместо блокирования доступа к общему ресурсу на время записи логов можно использовать асинхронные механизмы логирования, чтобы не блокировать основной поток выполнения.

Оптимизировать процесс логирования: если возможно, уменьшить объем информации, записываемый в логи, или оптимизировать сам процесс логирования для снижения его влияния на производительность.

1. Провести повторную итерацию нагрузочного тестирования после оптимизации. Это позволит подтвердить, что оптимизация проведена успешно и система удовлетворяет всем требованиям производительности.

# Методика тестирования

Нагрузочное тестирование проводилось в соответствии с документом Методика нагрузочного тестирования разработанного в компании «Перфоманс Лаб» и согласованным с Заказчиком.

## Профили нагрузки

В качестве исходных данных для формирования нагрузочного профиля был выбран период нагрузки системы с 1 декабря 2023г. до 31 декабря 2023г. Для наиболее часто используемых операций был выбран пиковый час нагрузки для каждой операции в отдельности (исходя из требования Заказчика).

Профиль нагрузки включает в себя операции регистрации пользователей, создания нового теста и загрузки видео проведения теста. Выполнение операций производится единовременно. Описание операций приведено в таблице 1.

Таблица 1 – Список операций нагрузочного профиля

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Название операции | Распределение % | Средняя нагрузка | Пиковая нагрузка в час |
| 1 | Создание нового теста | 41% | 1394 | 2787 |
| 2 | Загрузка видео | 24% | 794 | 1588 |
| 3 | Регистрация тестировщика | 24% | 792 | 1585 |
| 4 | Регистрация клиента | 11% | 385 | 770 |
|  | **Итого** | **100,0%** | **3365** | **6730** |

Предложенный в проекте метод подготовки нагрузочного профиля не является корректным, поскольку значение нагрузки для каждой операции было принято по максимальному значению в разные дни рассматриваемого периода. В дальнейшем нагрузочный профиль следует формировать следующим образом. Для правильного составления профиля для нагрузочного тестирования необходимо выбирать день с максимальной интенсивностью и пиковой нагрузкой, когда наиболее часто используемые и ресурсоемкие операции совместно создают наибольшую нагрузку.

## Эмуляция нагрузки

Моделирование нагрузки производится с использованием средств НТ Apache JMeter 5.6.3, путем эмуляции http-запросов. Генерируемые данные включают в себя информацию о:

* пользователях-тестировщиках/пользователях-клиентах;
* тестовых сценариях;
* видео проведенного теста.

Для подачи нагрузки используются нагрузочные скрипты. Все сценарии нагрузочных скриптов запускается одновременно.

При записи скриптов воспроизводятся статические элементы (.svg, .js, .css).

В процессе тестирования изменение суммарной интенсивности выполнения сценариев всеми виртуальными пользователями (моделируемая нагрузка на ИС) производилось путем изменения количества виртуальных пользователей, выполняющих сценарии, и величины задержки между последовательными итерациями.

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

## Критерии успешности проведения тестов

Тест считается успешным, если:

* в процессе тестирования запросы выполнялись с частотой, соответствующей профилю тестирования (в процессе тестирования возникло не более 5% ошибок);
* в процессе тестирования общая загрузка CPU составляет не более 70%;
* в процессе тестирования общая загрузка RAM составляет не более 80%;
* в процессе тестирования время выполнения каждого http - запроса составляет не более 5 сек.

## Проведённые тесты

В процессе проведения нагрузочного тестирования были проведены тесты согласно методике нагрузочного тестирования. Описания и результаты валидных тестов приведены в таблице ниже.

*Таблица 2 Перечень проведенных тестов*

|  |  |  |
| --- | --- | --- |
| Тест (Описание теста) | Используемые профили,  Длительность | Результаты |
| Определение максимальной производительности (Определение максимальных рабочих параметров системы и ее компонентов.) | 2%(от профиля Заказчика) + 2% на каждую ступень  Длительность: 10 мин выход на каждую ступень + 10 минут подтверждения ступени | Максимальная производительность достигнута при нагрузке 6% от текущего профиля нагрузки. При нагрузке 8% от текущего профиля нагрузки наблюдается резкий скачок по загрузки CPU. Наибольшую нагрузку на CPU создает сценарий создания теста (UC\_03\_new\_test\_creating).  [Метрики теста](https://snapshots.raintank.io/dashboard/snapshot/3IPwY9bo9CCPFTopq3xEWcS1XLACQCL3) [Системные метрики](https://snapshots.raintank.io/dashboard/snapshot/4UfOB9UrL7qqfu3g7a2qb0Bywy389nDa) [Метрики попроцессно](https://snapshots.raintank.io/dashboard/snapshot/X65FwipT0CivwMJSuWgsJpxO3xwu1lrW) [Метрики БД](https://snapshots.raintank.io/dashboard/snapshot/PHB5iHkTNiR7ch1Di8k2baQ8DGUHfGSx) [Метрики БД - 2](https://snapshots.raintank.io/dashboard/snapshot/Tdu8XYfYhbtyppy4De1GNQK3VioMWM0v) |
| Подтверждение максимальной производительности (Определение способности системы поддерживать высокий уровень производительности.) | 100% (6% от профиля Заказчика) | При выбранном уровне нагрузки критерии успешности были  нарушены. Загрузка CPU составила 72,18%  [Метрики теста](https://snapshots.raintank.io/dashboard/snapshot/v80F3Fe9CHqhHf1oRjjk2GKKkOOYw1u0)  [Системные метрики](https://snapshots.raintank.io/dashboard/snapshot/DU6KGL9IqmQlM6GbRbNpprG1bSyuFw8u) |
| Подтверждение максимальной производительности (Определение способности системы поддерживать высокий уровень производительности.) | 100% (4% от профиля Заказчика)  Длительность: 60 мин | При выбранном уровне нагрузки критерии успешности нарушены не были.  [Метрики теста](https://snapshots.raintank.io/dashboard/snapshot/RuodkOsgGA5ipUUXSktYCZj0B9WnQ806) [Системные метрики](https://snapshots.raintank.io/dashboard/snapshot/jiUqB87U63GhVD5bPHD3X3npNRb8At4d) [Метрики попроцессно](https://snapshots.raintank.io/dashboard/snapshot/N675cVVNSTvm0vYnrveEdHELjyFPHJk9) [Метрики БД](https://snapshots.raintank.io/dashboard/snapshot/EcIonGxQb51YexayS1DWTsRh216B2bRn) [Метрики БД - 2](https://snapshots.raintank.io/dashboard/snapshot/vGsGIx9x5bp0CCi80qMvGdK5r0UQWToC) |
| Тестирование c увеличением нагрузки до  150% от найденного максимума  (Оценка реакции системы на увеличение нагрузки сверх предельного значения.) | 75% -> 150% -> 75% от найденного максимума (от 4% профиля Заказчика)  Длительность: 60 мин (  15 мин + 15 мин + 30 мин) | Система за отведенное время теста после возвращения на нагрузку 75% от найденного максимума не вернулась в рамки критериев успешности по временам отклика. Запрос /api/tester/create-task?orderId=10581351 выходил за пределы 5 секунд до остановки тестирования.  [Метрики теста](https://snapshots.raintank.io/dashboard/snapshot/LmHTeWJfbjieJVNYMX00cCEMhurk6tOv) [Системные метрики](https://snapshots.raintank.io/dashboard/snapshot/ph0I5WwWLysvKJvyaDaGL3r3uMvNBPeZ) [Метрики попроцессно](https://snapshots.raintank.io/dashboard/snapshot/fTXyOBlzjyJTPYNW9EY8EArt78Epm98L)  [Метрики БД](https://snapshots.raintank.io/dashboard/snapshot/mmHnGAe1fuAEQjW24qMjoiJTckrF54Ct) [Метрики БД - 2](https://snapshots.raintank.io/dashboard/snapshot/Pna7yyf02ceAtXNqq1TPLcVlATZenkwo) |
| Тестирование при отказе БД  (Оценка работоспособности системы при отказе работы БД.) | 75% от найденного максимума (от 4% профиля Заказчика)  Длительность: 60 мин (  15 мин + 15 мин (отключение) + 30 мин) | Система за время теста после включения БД не вернулась в рамки критериев успешности по временам отклика запросов, которые превышали пределы 5 секунд до остановки тестирования (перечень запросов см. рисунок 20). Время отклика запросов составляло 30 секунд.  [Метрики теста](https://snapshots.raintank.io/dashboard/snapshot/JYSEQGsaHg9DNCgwryE3QF5pWnsMSXbR) [Системные метрики](https://snapshots.raintank.io/dashboard/snapshot/SDyexzv1xRuMCjxx7xrDTkb7T1geFT63) [Метрики попроцессно](https://snapshots.raintank.io/dashboard/snapshot/LvBL0dTmU36O0hHJpyIIrXKamt1657sk) [Метрики БД](https://snapshots.raintank.io/dashboard/snapshot/nzkranEdngP4SqxtubJeqBt8YVWUWozQ) [Метрики БД - 2](https://snapshots.raintank.io/dashboard/snapshot/dicAMtd918iRoFhHA6I7eMK1aq1K7qY0) |
| Тестирование сбоя  (Оценка работоспособности и отказоустойчивости системы после ее сбоя.) | 75% от найденного максимума (от 4% профиля Заказчика)  Длительность: 60 мин (  15 мин + 15 мин (отключение) + 30 мин) | После восстановления система вернулась в рамки критериев успешности по времени отклика запроса (api/tester/create-task?orderId=105) за 7 минут. Остальные запросы за рамки критерием успешности не выходили.  [Метрики теста](https://snapshots.raintank.io/dashboard/snapshot/skaBDldTm7XSAQspZKvvhtRW9ch8sAYy) [Системные метрики](https://snapshots.raintank.io/dashboard/snapshot/qiHURVuHVBhpNuPgtzh0X6wp34wjF6H1) [Метрики попроцессно](https://snapshots.raintank.io/dashboard/snapshot/iGHeykJuU2J3afjmLnJnbbzx3CsmN3kP) [Метрики БД](https://snapshots.raintank.io/dashboard/snapshot/Dg2s7ozxfBJ0ZLjmx99sFlH1Maes6Qm2) [Метрики БД - 2](https://snapshots.raintank.io/dashboard/snapshot/9MozktMlbdrx9ay6aDTNCx4JgetGXu4r) |

## 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.
* Для прохождения успешного прохождения некоторых тестов будут внесены изменения в БД.
* Тестирование не направлено на выявление дефектов в аппаратной части стенда.
* Перед проведением тестирования на этапе создания нагрузочных скриптов версии компонентов информационной системы фиксируются и не изменяются до окончания тестирования, за исключением случаев устранения ошибок, мешающих дальнейшему проведению работ по тестированию.
* Тестирование производится с учетом того, что со стороны Заказчика не предоставлены доступы и контакты серверов продуктивного стенда.
* Для корректного поведения сценариев нагрузочного тестирования в них вносятся изменения, позволяющие миновать ошибки со стороны тестового стенда путем прямого взаимодействия с базой данных.
* После проведения оптимизации базы данных, требуется сохранить ее в едином состоянии для последующего анализа.
* Объем данных в БД на котором производится тестирование не соответствует промышленному стенду из-за отсутствия доступа к промышленному серверу. Перед началом тестирования была проведена частичная чистка таблиц.

# Результаты нагрузочного тестирования

В данном разделе представлены следующие результаты нагрузочных тестов

* поиск максимальной производительности;
* подтверждение максимальной производительности
* тестирование отказоустойчивости системы;
* тестирование надежности системы.

## Результаты проведения теста поиска максимальной производительности

В тесте на определение максимальной производительности выполнено постепенное повышение нагрузки с шагом 2% от профиля нагрузки Заказчика. При выходе на новую ступень нагрузки для каждого сценария добавляется по одному виртуальному пользователю.

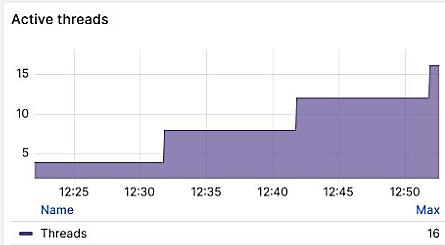


Рисунок 1 - Количество активных виртуальных пользователей

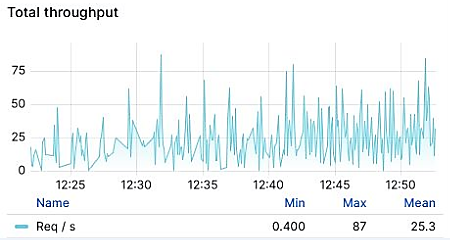


Рисунок 2 - Количество запросов в секунду

На рисунках 2 и 3 видна зависимость количества запросов в секунду от количества виртуальных пользователей.

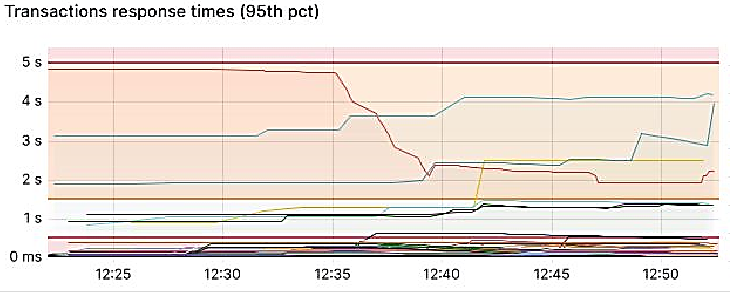




Рисунок 3 - Время отклика запросов на 95 pct

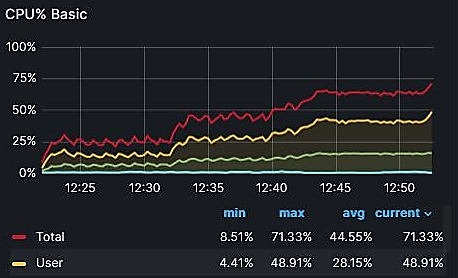


Рисунок 4 - График загрузк CPU

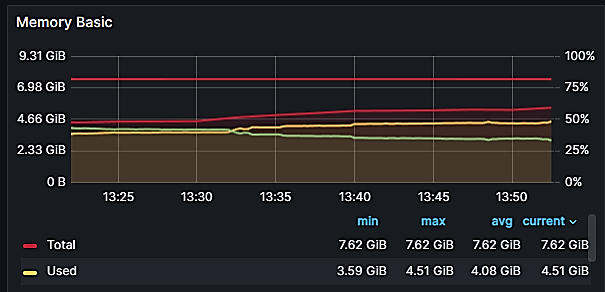


Рисунок 5 - График загрузки оперативной памяти

Достигнута максимальная производительность системы при нагрузке в 6% от текущего профиля с 12 пользователями. Однако, когда нагрузка увеличивается до 8% (16 VU), наблюдается резкий скачок по загрузке процессора. В течение тестирования также наблюдается увеличение использования оперативной памяти.

## Результаты проведения теста подтверждения максимальной производительности

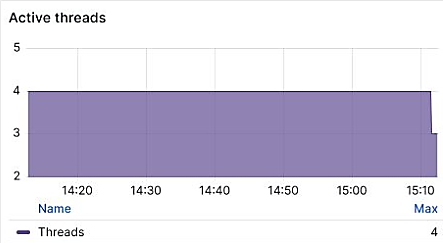


Рисунок 6 - Количество активных виртуальных пользователей

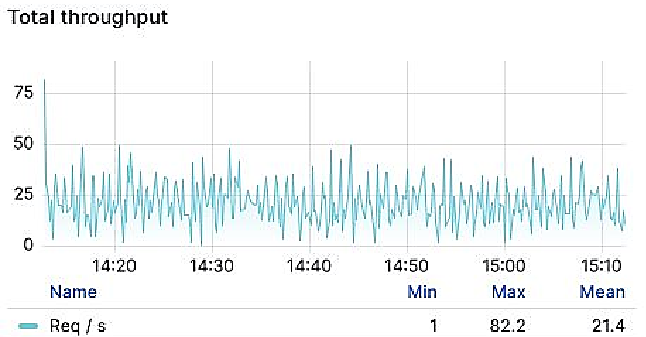
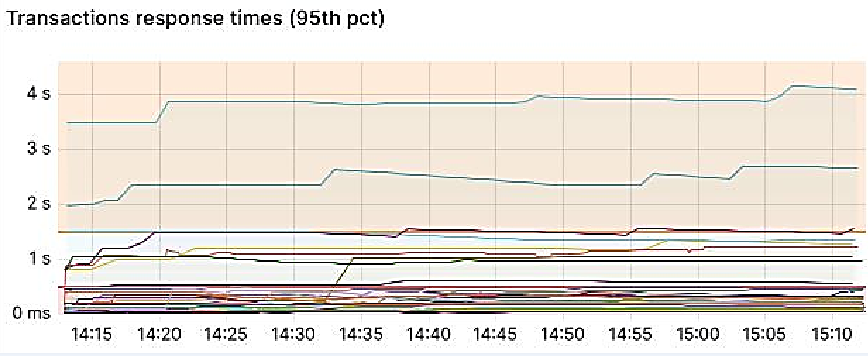


Рисунок 7 - Количество запросов в секунду



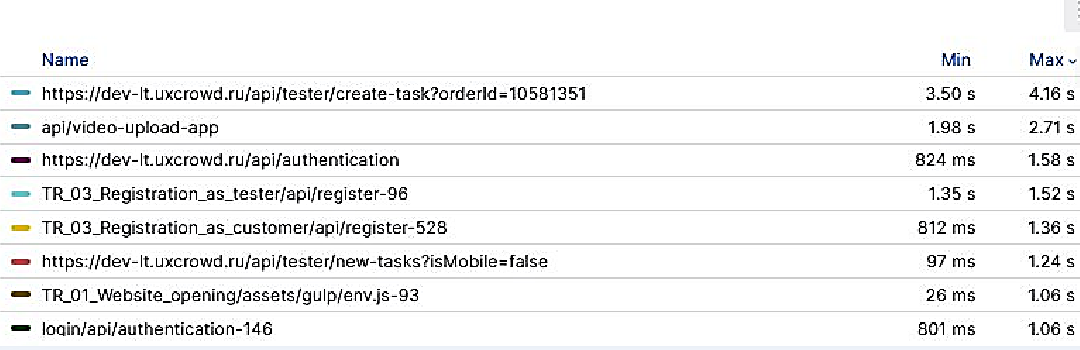


Рисунок 8 - Время отклика запросов на 95 pct

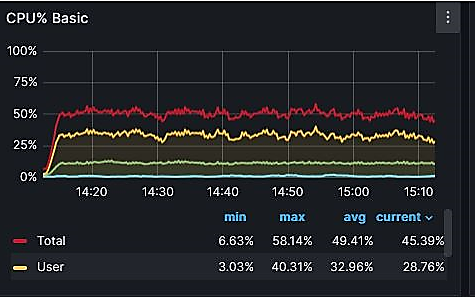


Рисунок 9 - График загрузки CPU



Рисунок 10 - График загрузки оперативной памяти

Для проведения теста подтверждения максимума принята нагрузка 4% от текущего профиля нагрузки. При выбранном уровне нагрузки критерии успешности нарушены не были.

## Результаты проведения теста с увеличением нагрузки до 150% от найденного максимума



Рисунок 11 - Количество активных виртуальных пользователей

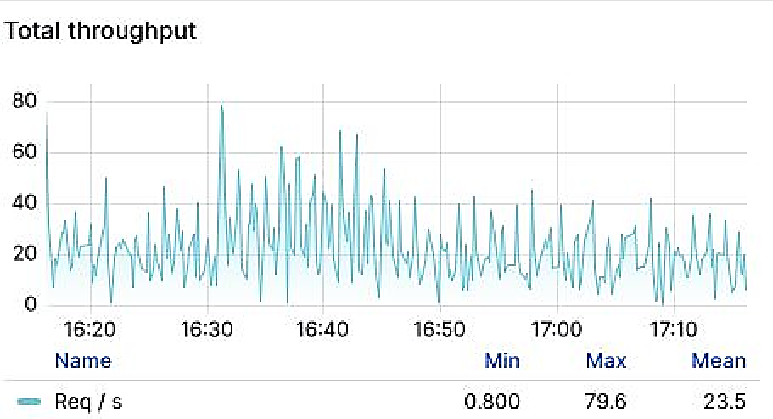
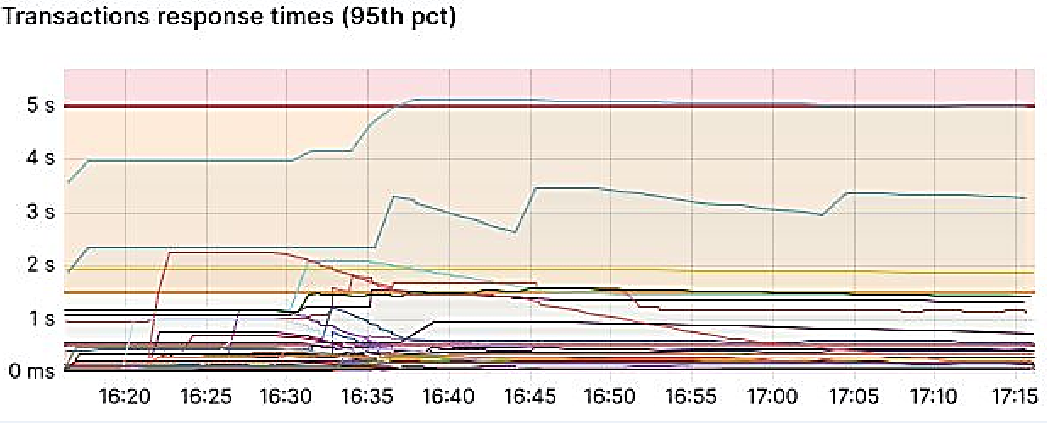


Рисунок 12 - Количество запросов в секунду



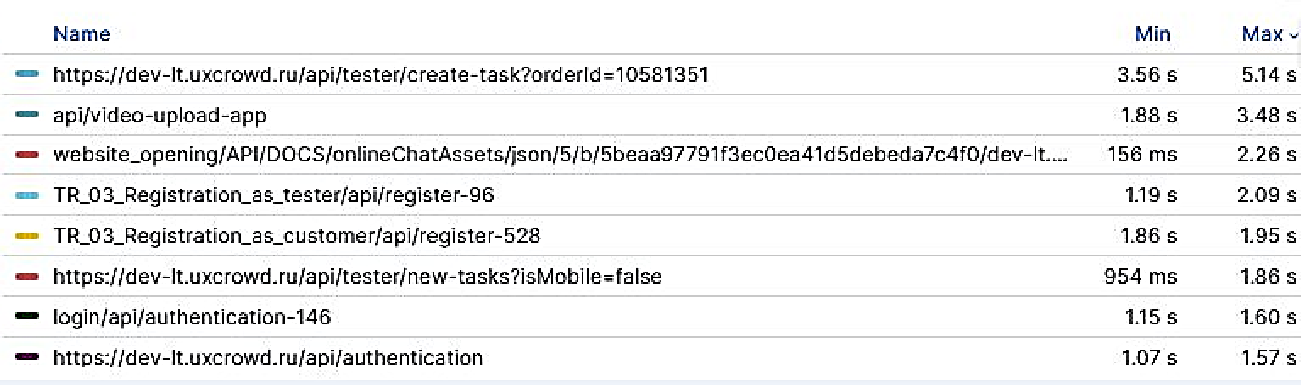


Рисунок 13 - Время отклика запросов на 95 pct

Система за отведенное время теста после возвращения на нагрузку 75% от найденного максимума не вернулась в рамки критериев успешности по временам отклика. Запрос /api/tester/create-task?orderId=10581351 выходил за пределы 5 секунд до остановки тестирования.

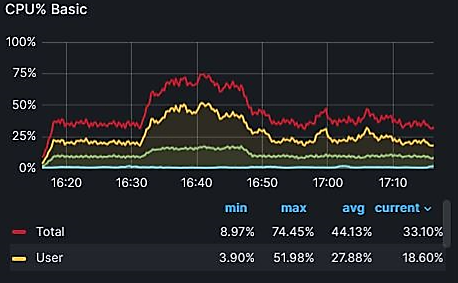


Рисунок 14 - График загрузки CPU

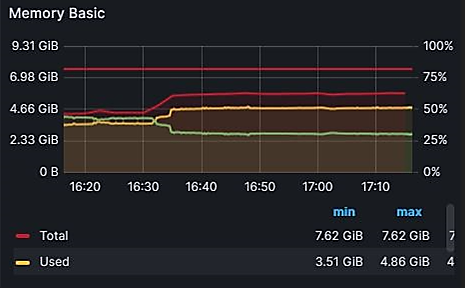


Рисунок 15 - График загрузки оперативной памяти

Как видно на рисунках 14, 15 после скачка интенсивности в 2 раза система демонстрировала нестабильность работы CPU и увеличенное потребление оперативной памяти даже после возвращения к 75% нагрузки.

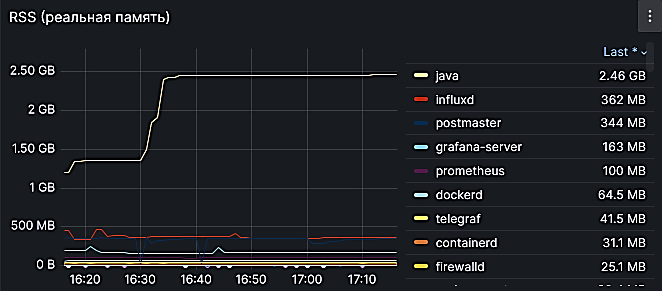


Рисунок 16 - Утилизация памяти дискового пространства пооперационно

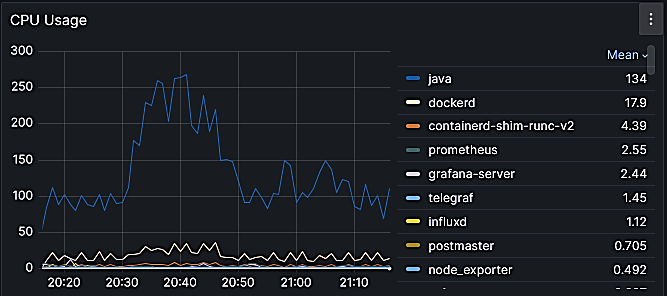


Рисунок 17 - Утилизация CPU попроцессно

В процессе работы системы наибольшее потребление CPU и дискового пространства  приходится на java-приложение. Причиной тому является утечка памяти. Подробное описание проблемы представлено в п. 8. Рекомендации по решению проблемы представлены в разделе 9.

## Результаты проведения теста при отказе БД

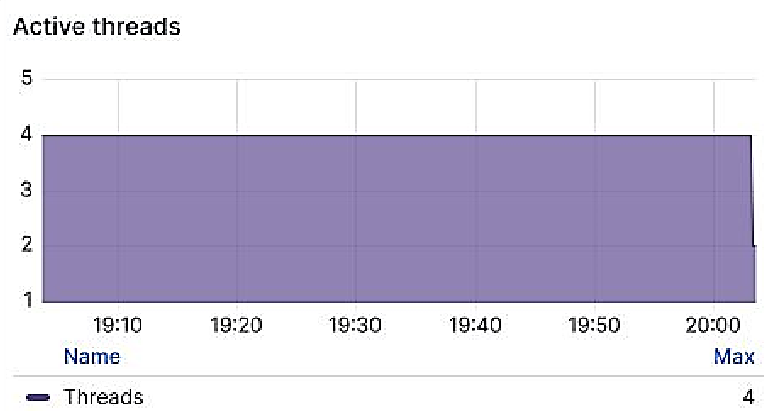


Рисунок 18 - Количество активных виртуальных пользователей

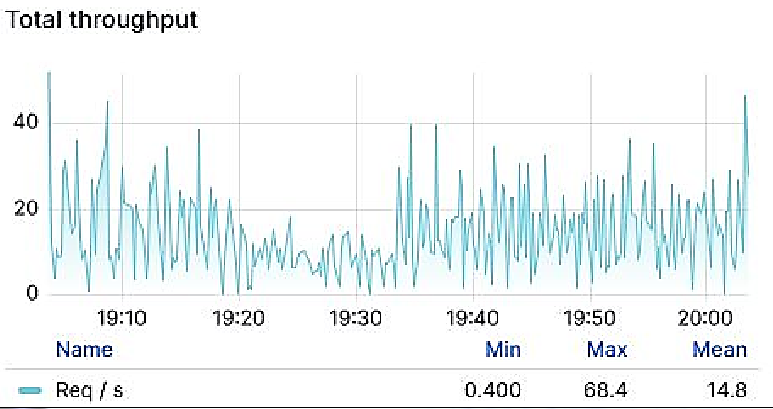
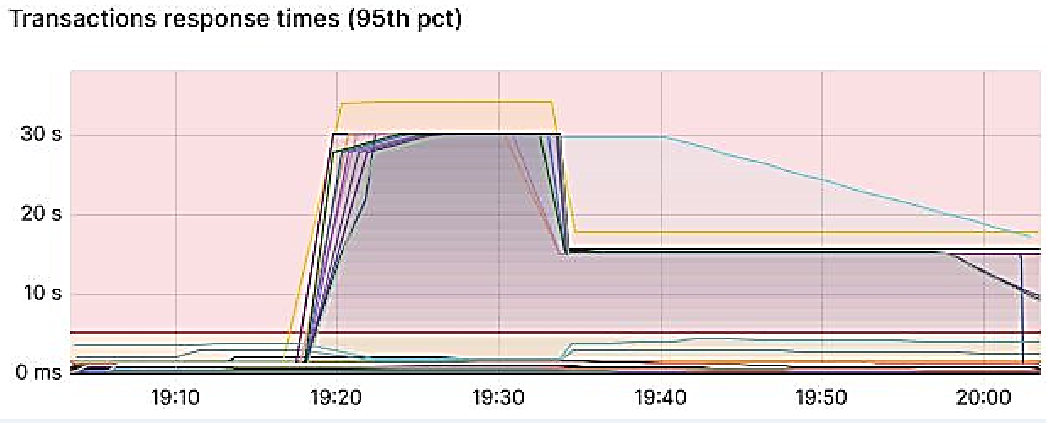


Рисунок 19 - Количество запросов в секунду



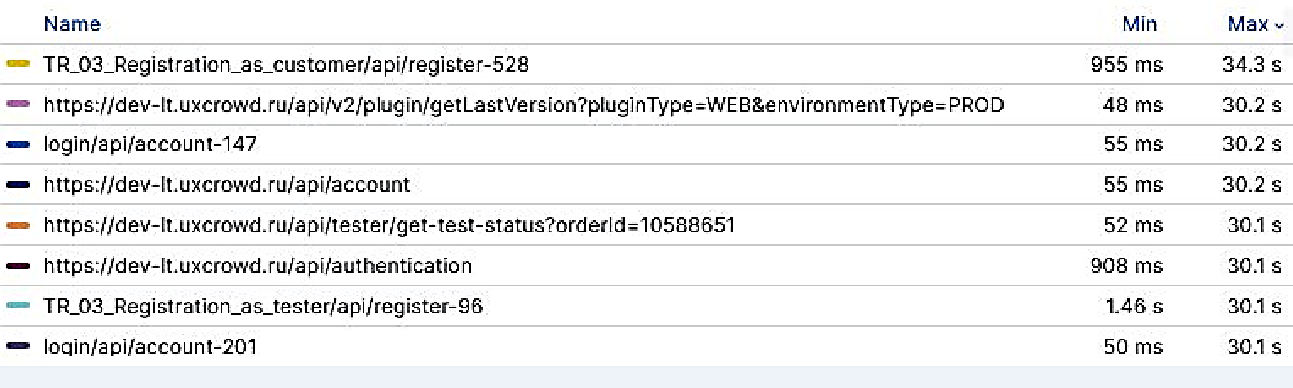


Рисунок 20 - Время отклика запросов на 95 pct

Система за отведенное время теста после включения БД не вернулась в рамки критериев успешности по временам отклика запросов, которые превышали пределы 5 секунд и не демонстрировали положительную тенденцию до остановки тестирования (30 секунд).

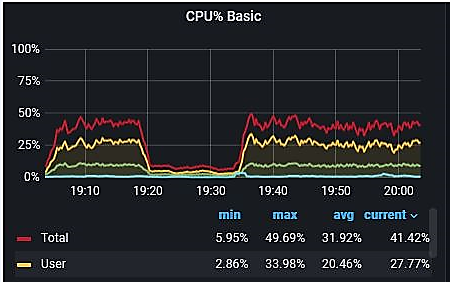


Рисунок 21 - График загрузки CPU

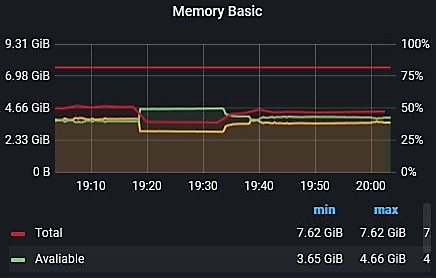


Рисунок 22 - График загрузки оперативной памяти

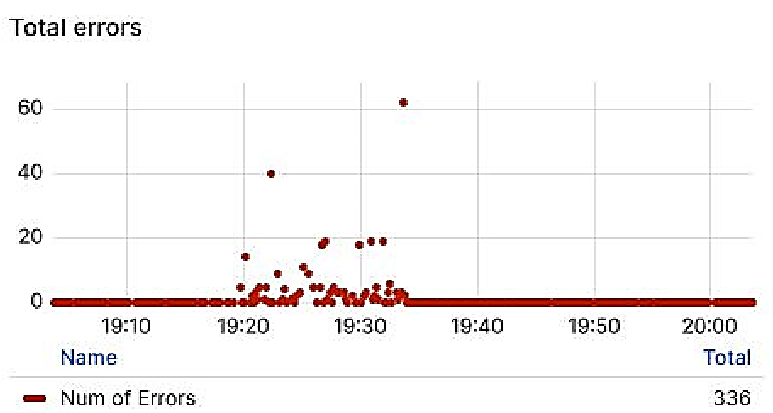


Рисунок 23- Количество ошибок

По результатам проведения теста при отказе БД было установлено (см. рисунок 23), что при отключении базы данных возникали ошибки, в то время как в остальных случаях процесс тестирования проходил успешно.

## Результаты проведения теста сбоя приложения

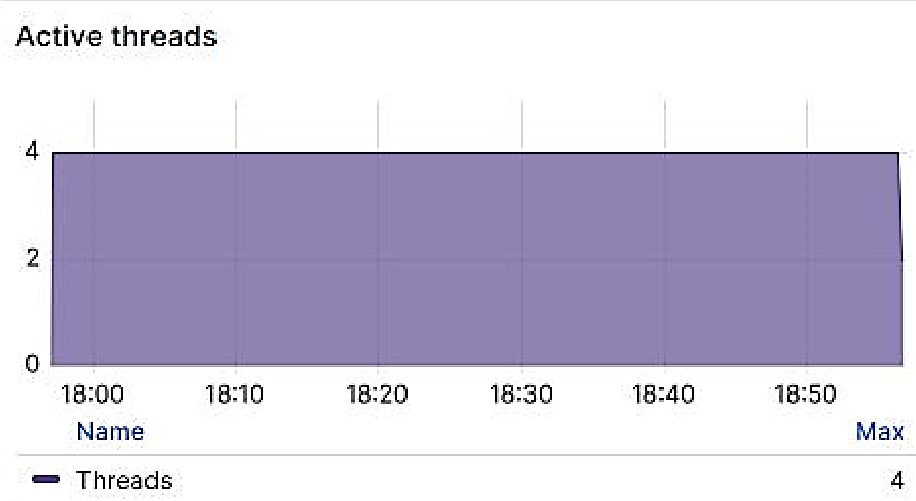


Рисунок 24 - Количество активных виртуальных пользователей

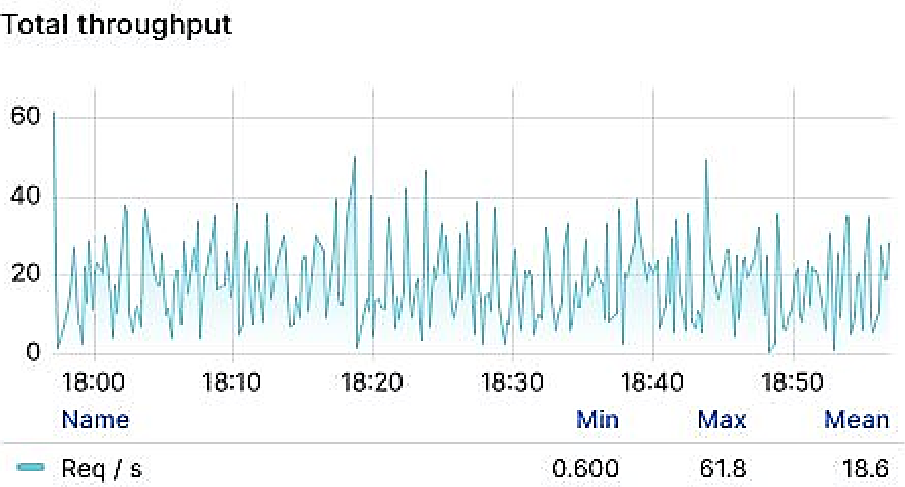
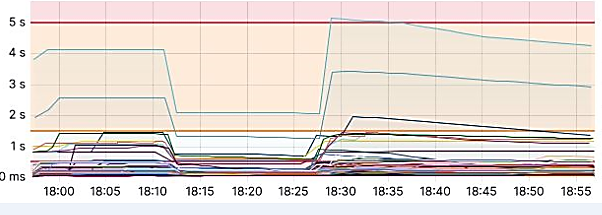


Рисунок 25 - Количество запросов в секунду



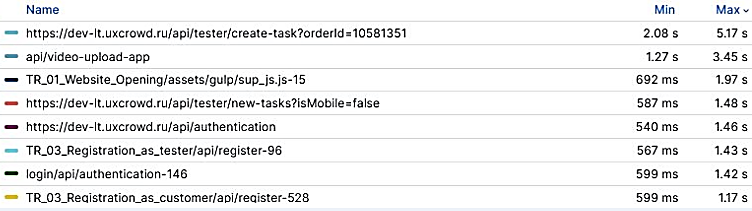


Рисунок 26 - Время отклика запросов на 95 pct

После восстановления работоспособности системы было зафиксировано, что время отклика запроса (api/tester/create-task?orderId=105) сократилось до установленных критериев успешности за 7 минут, причем все остальные запросы продолжали укладываться в заданные временные рамки.

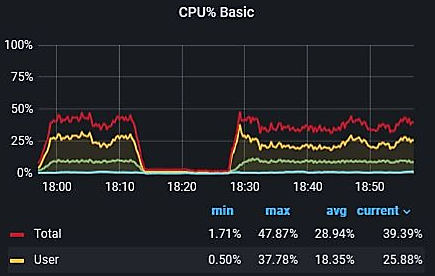


Рисунок 27 - График загрузки CPU

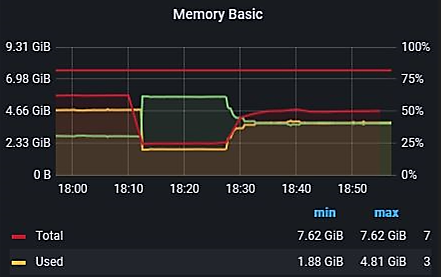


Рисунок 28 - График загрузки оперативной памяти

Как можно заметить из представленных графиков, после восстановления работы приложения произошло заметное снижение нагрузки на систему, что отразилось на показателях использования процессора и оперативной памяти, демонстрируя их уменьшение.

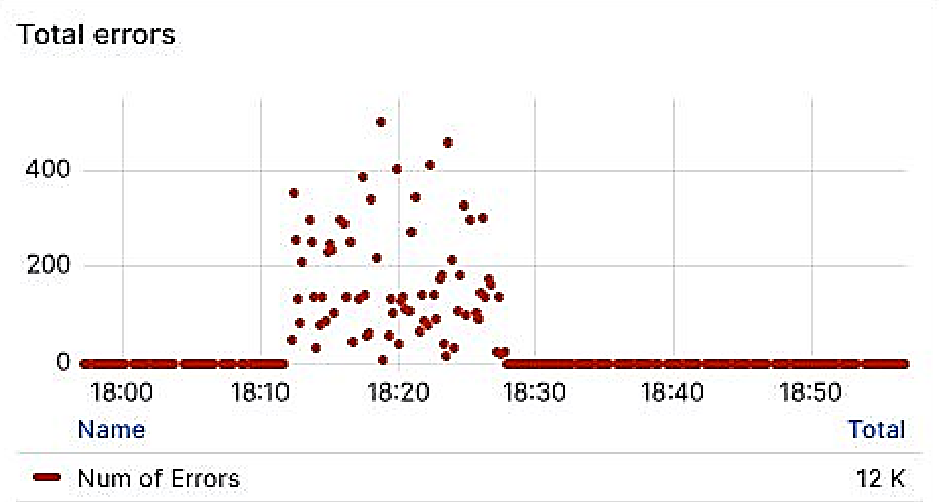


Рисунок 29 - Количество ошибок

Из рисунка выше видно, что ошибки проявлялись исключительно при сбое приложения, в то время как в остальных случаях все проходило успешно.

## Результаты проведения теста определения надежности

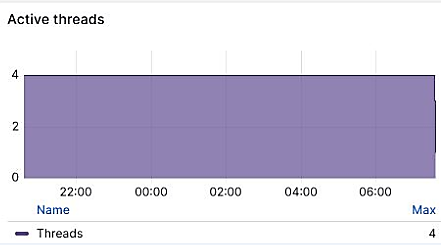


Рисунок 30 - Количество активных виртуальных пользователей

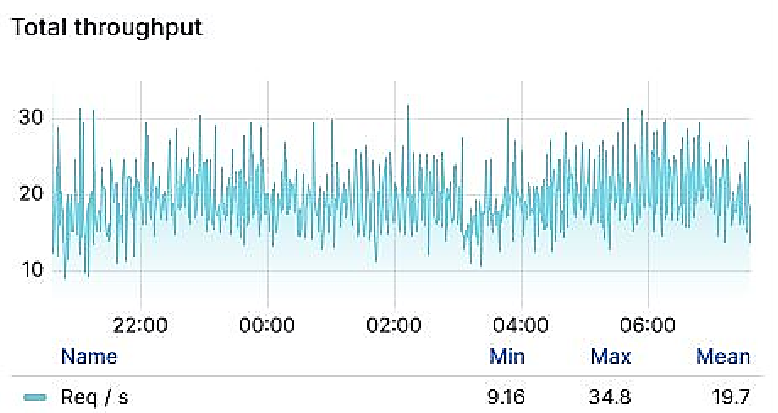
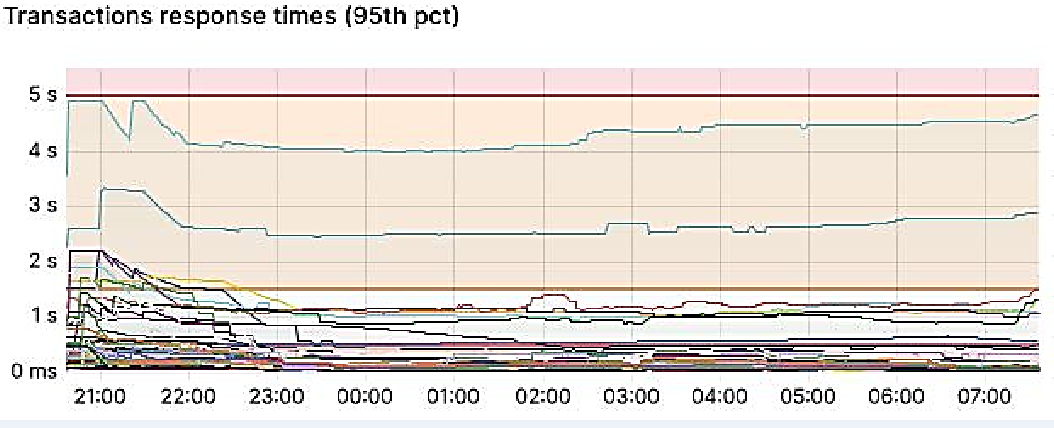


Рисунок 31 - Количество запросов в секунду



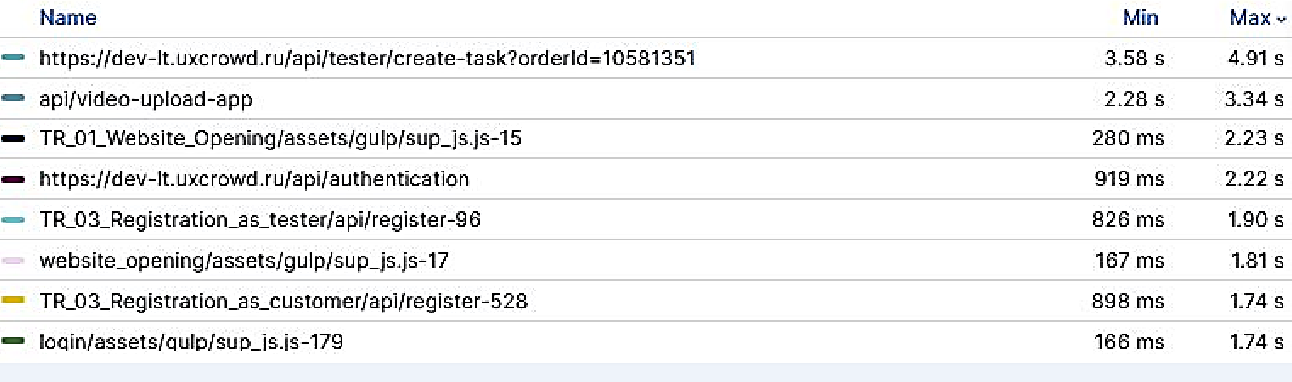


Рисунок 32 - Время отклика запросов на 95 pct

В процессе проведения теста на стабильность было замечено, что время отклика запросов постепенно увеличивалось. Например, время отклика на запрос /api/tester/create-task?orderId=10581351 увеличилось с 3.99 до 4.68 секунды, что свидетельствует о потенциальных проблемах с производительностью системы и необходимости проведения дополнительных оптимизаций.

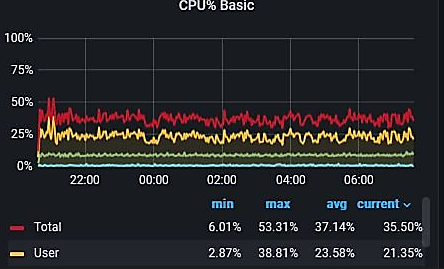


Рисунок 33 - График загрузки CPU

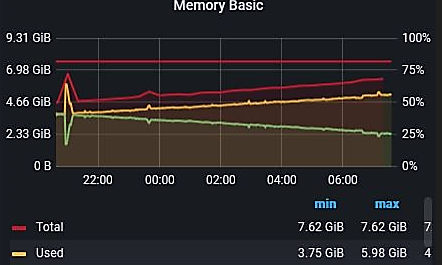


Рисунок 34 - График загрузки оперативной памяти

В ходе тестирования стабильности не было замечено нарушений критериев успешности. Однако на рисунке 34 наблюдается устойчивая тенденция увеличения использования оперативной памяти, что может являться сигналом об утечке памяти.

# Приложение

Приложение А

select distinct customeror0\_.id as id1\_22\_, customeror0\_1\_.created as created2\_22\_, customeror0\_1\_.updated as updated3\_22\_, customeror0\_1\_.version as version4\_22\_, customeror0\_1\_.customer\_id as custome13\_22\_, customeror0\_1\_.introduction

as introduc5\_22\_, customeror0\_1\_.pre\_screener\_intro as pre\_scre6\_22\_, customeror0\_1\_.pre\_screener\_outro as pre\_scre7\_22\_, customeror0\_1\_.start\_date as start\_da8\_22\_, customeror0\_1\_.status as status9\_22\_, customeror0\_1\_.target\_site

as target\_10\_22\_, customeror0\_1\_.tariff\_plan\_id as tariff\_14\_22\_, customeror0\_1\_.test\_type as test\_ty11\_22\_, customeror0\_1\_.orders\_type as orders\_12\_22\_, customeror0\_.app\_name as app\_name1\_8\_, customeror0\_.consent\_document\_id

as consent\_9\_8\_, customeror0\_.payment as payment10\_8\_, customeror0\_.test\_title as test\_tit2\_8\_, customeror0\_.tester\_type as tester\_t3\_8\_, customeror0\_.their\_fc\_testers\_count as their\_fc4\_8\_, customeror0\_.their\_testers\_count

as their\_te5\_8\_, customeror0\_.url\_completed\_order as url\_comp6\_8\_, customeror0\_.url\_completed\_task as url\_comp7\_8\_ from customer\_orders customeror0\_ inner join orders customeror0\_1\_ on customeror0\_.id=customeror0\_1\_.id

cross join customer customer24\_ inner join users customer24\_1\_ on customer24\_.id=customer24\_1\_.id where customeror0\_1\_.customer\_id=customer24\_.id and (customeror0\_.id in (select customeror1\_.id from customer\_orders customeror1\_

inner join orders customeror1\_1\_ on customeror1\_.id=customeror1\_1\_.id where (select count(distinct testertask3\_.id) from customer\_orders customeror2\_ inner join tester\_task testertask3\_ on customeror2\_.id=testertask3\_.orders\_id

where (testertask3\_.pre\_screener\_status='A' or testertask3\_.pre\_screener\_status='B') and testertask3\_.tester\_id<>401357 and testertask3\_.pre\_screener\_created>='2024/12/12'

and customeror1\_.id=customeror2\_.id)<((select sum(cast(participan5\_.count as int8)) from customer\_orders customeror4\_ inner join participant\_group participan5\_ on customeror4\_.id=participan5\_.order\_id

where customeror1\_.id=customeror4\_.id)-(select count(distinct testertask7\_.id) from customer\_orders customeror6\_ inner join tester\_task testertask7\_ on customeror6\_.id=testertask7\_.orders\_id where (testertask7\_.status in ('COMPLETED', 'CANCELLED'))

and customeror1\_.id=customeror6\_.id))\*2) or exists (select testertask8\_.id from tester\_task testertask8\_ where testertask8\_.pre\_screener\_created>='2024/12/12' and (testertask8\_.pre\_screener\_status='D' or testertask8\_.pre\_screener\_status='E')

and testertask8\_.tester\_id=401357)) and (customeror0\_.id not in  (select customeror9\_.id from customer\_orders customeror9\_ inner join orders customeror9\_1\_ on customeror9\_.id=customeror9\_1\_.id inner join tester\_task testertask10\_

on customeror9\_.id=testertask10\_.orders\_id where testertask10\_.tester\_id=401357 and (testertask10\_.pre\_screener\_status is not null) and (testertask10\_.pre\_screener\_status='J' or testertask10\_.pre\_screener\_status='F'

and (testertask10\_.status in ('COMPLETED', 'CANCELLED'))))) and (select count(customeror11\_.id) from customer\_orders customeror11\_ inner join orders customeror11\_1\_ on customeror11\_.id=customeror11\_1\_.id inner join participant\_group participan12\_

on customeror11\_.id=participan12\_.order\_id cross join customer customer14\_ where customeror11\_1\_.customer\_id=customer14\_.id and participan12\_.count>cast((select count(distinct testertask13\_.id) from tester\_task testertask13\_

where testertask13\_.participant\_group\_id=participan12\_.id and testertask13\_.status<>'COMPLETED' and testertask13\_.status<>'CANCELLED' and (testertask13\_.status<>'COMPLETED' or testertask13\_.tester\_id<>401357)) as int4)

and customeror0\_.id=participan12\_.order\_id and (customer14\_.block\_past\_tester='f' or customeror11\_1\_.customer\_id not in  (select pasttester15\_.customer\_id from past\_tester\_lock pasttester15\_ cross join customer customer16\_

inner join users customer16\_1\_ on customer16\_.id=customer16\_1\_.id where pasttester15\_.customer\_id=customer16\_.id and pasttester15\_.tester\_id=401357 and customer16\_.block\_past\_tester='f')) and (participan12\_.gender='ANY'

or participan12\_.gender='ANY') and (participan12\_.family\_status='ANY' or participan12\_.family\_status='ANY') and (participan12\_.childrens='ANY' or participan12\_.childrens='ANY') and (401357 in (select socialstat18\_.social\_status\_id

from part\_group\_social\_statuses socialstat18\_ where participan12\_.id=socialstat18\_.participant\_group\_id) or 401357 in (select socialstat19\_.social\_status\_id from part\_group\_social\_statuses socialstat19\_

where participan12\_.id=socialstat19\_.participant\_group\_id) or (select count(socialstat25\_.participant\_group\_id) from part\_group\_social\_statuses socialstat25\_ where participan12\_.id = socialstat25\_.participant\_group\_id)=95831)

and (2 in (select educations20\_.education\_id from part\_group\_educations educations20\_ where participan12\_.id=educations20\_.participant\_group\_id) or 2 in (select educations21\_.education\_id from part\_group\_educations educations21\_

where participan12\_.id=educations21\_.participant\_group\_id) or (select count(educations26\_.participant\_group\_id) from part\_group\_educations educations26\_ where participan12\_.id = educations26\_.participant\_group\_id)=95831)

and (participan12\_.min\_age<=18 and participan12\_.max\_age>=61 or cast(20 as int8)<=61 and participan12\_.max\_age>=18) and (participan12\_.household\_income\_min<=0 and participan12\_.household\_income\_max>=55000

or participan12\_.household\_income\_max>0 and participan12\_.household\_income\_max=20000))<>55000 and  not (exists (select customeror22\_.id from customer\_orders customeror22\_ inner join orders customeror22\_1\_

on customeror22\_.id=customeror22\_1\_.id inner join tester\_task testertask23\_ on customeror22\_.id=testertask23\_.orders\_id where customeror0\_.id=testertask23\_.orders\_id and testertask23\_.tester\_id=401357 and (testertask23\_.status

in ('COMPLETED', 'CANCELLED')))) and customeror0\_.tester\_type='OUR' and customer24\_1\_.locale='ru\_RU' and customeror0\_1\_.status='ACTIVE'

# 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:

Минин Владимир Евгеньевич (minin.vladimir.90@gmail.com, 8-925-658-12-05)

Гайбович Алексей Николаевич (alexyng7@gmail.com, 8-922-684-43-21)

Нельзин Илья Юрьевич (illobikol@gmail.com, 8-982-990-76-72)