# Цели тестирования

|  |  |  |
| --- | --- | --- |
| Цель тестирования | Вид тестирования | DOD (Definition of done) |
| Определить максимальную производительность системы, оценить запас производительности ИС на ПРОД, определить узкое место системы | Поиск максимальной производительности | В ходе теста поиска максимума достигнута точка деградации;  На протяжении интервала стабильной подачи нагрузки теста подтверждения система соответствовала требованиям;  В выводах протокола НТ зафиксирована МП как в % от профиля нагрузки, так и в % запаса производительности относительно промышленной нагрузки; также указывается ограничитель производительности, т.е. причина отказа системы при достижении точки деградации (узкое место) |
| Подтвердить способность системы автоматически восстанавливаться в стрессовых ситуациях | Стресс-тестирование (все сценарии) | Все сценарии стресс-тестирования пройдены с соответствием критериев успешности для каждого сценария |

# Ограничения тестирования

|  |  |  |
| --- | --- | --- |
| Ограничение | Влияние | Срок устранения / задача на устранение |
| Отсутствие прикладного мониторинга | Невозможность диагностики проблем производительности на уровне внутренних механизмов | 2Q2025 |
| В качестве тестовых данных используются синтетически сгенерированные данные | Возможное отклонение вариативности тестовых данных от ПРОМ среды | 2Q2025 |

# Объект тестирования

WebTours 1.0 — это учебное веб-приложение, разработанное для демонстрации возможностей инструментов нагрузочного тестирования, таких как LoadRunner. Оно представляет собой онлайн-агентство путешествий, предоставляющее пользователям функциональность для бронирования авиаперелетов.

## Компоненты объекта тестирования

* **Фронтенд (Frontend):** Интерфейс пользователя реализован с использованием стандартных веб-технологий: HTML, CSS и JavaScript. Эти технологии обеспечивают отображение веб-страниц и взаимодействие с пользователем.​
* **Бэкенд (Backend):** Серверная часть приложения отвечает за обработку бизнес-логики и взаимодействие с базой данных. WebTours использует язык программирования Perl, а именно дистрибутив Strawberry Perl, который позволяет запускать Perl-скрипты на Windows-платформах.​
* **База данных (Database):** Для хранения данных о пользователях и бронированиях используется простая файловая система в виде текстовых файлов.​

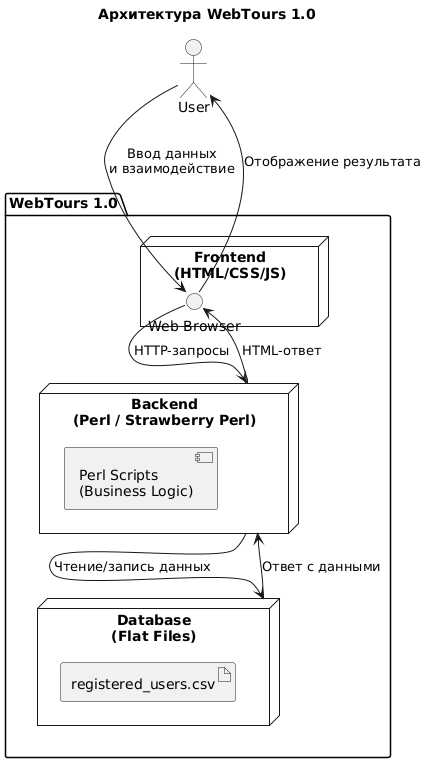


Рисунок – Архитектура WebTours

## Архитектура стенда НТ

|  |  |
| --- | --- |
| Ресурс | Значение |
| CPU | 0,5 |
| RAM | 0,5 gb |

WebTours следует классической трехзвенной архитектуре, состоящей из следующих уровней:​

1. **Клиентский уровень (Client Tier):** Включает веб-браузеры пользователей, которые отправляют HTTP-запросы и отображают полученные страницы.​
2. **Сервер приложений (Application Server):** Обрабатывает запросы от клиентов, выполняет бизнес-логику и взаимодействует с базой данных. Реализован с использованием Perl-скриптов, работающих на сервере.​
3. **Уровень данных (Data Tier):** Состоит из текстовых файлов, используемых для хранения данных приложения.

## Требования к производительности

### Критерии оценки работоспособности системы

Система должна иметь запас 200% от ПРОМа (100% + 200%).

## Мониторинг (обновить ссылки)

|  |  |  |
| --- | --- | --- |
| Назначение мониторинга | Инструмент | Ссылка |
| Мониторинг метрик инструмента НТ (JMeter/Gatling) | JMeter/Gatling (с подключением к Grafana) | http://localhost:3000 (Grafana) |
| Мониторинг утилизации ресурсов контейнеров (Docker) | Grafana (стандартный дашборд для Docker) | http://localhost:3000 (Grafana) |
| Сбор метрик контейнеров (cAdvisor) | cAdvisor | http://localhost:8080 |
| Хранение и обработка метрик (Prometheus) | Prometheus | http://localhost:9090 |
| Визуализация метрик (Grafana) | Grafana | http://localhost:3000 |

### Дополнительные средства мониторинга

# Моделирование нагрузки

## Виды тестирования

### Поиск максимальной производительности

 Тест проводится с **пошаговым увеличением нагрузки** для выявления предельного уровня производительности системы.

 Каждая ступень нагрузки увеличивается на **100% от базового профиля** предыдущей ступени.

 **Длительность стабильного отрезка ступени** – **22 минуты** (включая 2 минуты для стабилизации и 20 минут для анализа).

 Завершается тест при достижении точки деградации системы.

### Подтверждение максимальной производительности

### Проводится на **100% от уровня максимальной производительности**, выявленного в тесте поиска.

* **Длительность стабильного отрезка ступени** – **1 час 2 минуты** (включая 2 минуты для стабилизации и 20 минут для анализа).

 **Цель:** Подтвердить работоспособность системы при длительной нагрузке без деградации ключевых метрик.

### Стресс-тест Пиковая производительность (где деградация началась)

 **Метод нагрузки:** Резкое повышение нагрузки до уровня, превышающего **пиковую производительность системы**.

 **Длительность удержания пиковой нагрузки:** **5 минут после фиксации деградации**.

 **Фаза спада нагрузки:** Плавное снижение нагрузки по ступеням с шагом **-25% от пиковой нагрузки**. (выстроить так сценарий, несколько миунт)

 **Длительность каждой ступени в фазе снижения:** **2 минуты**.

 **Цель:** Проверка поведения системы в условиях экстремальной нагрузки и восстановления после перегрузки.

### Стресс-тест объёмное тестирование

 **Метод нагрузки:** Для **30% пользователей** **увеличивается объем данных в 10 раз** (увеличение количества купленных билетов).

### Проводится на **100% от уровня максимальной производительности**, выявленного в тесте поиска.

* **Длительность стабильного отрезка ступени** – **1 час 2 минуты** (включая 2 минуты для стабилизации и 20 минут для анализа).

 **Цель:** Оценка влияния увеличенного объема данных на производительность и корректность обработки операций.

 **Критерии успешности:** Отсутствие значительной деградации времени отклика и корректное выполнение операций.

## Профиль нагрузки

Таблица – Профиль нагрузки

|  |  |  |  |
| --- | --- | --- | --- |
| Статистика с ПРОДа | | Профиль | |
| Название запроса | Интенсивность по статистике запросов / час | Расчетная интенсивность запросов / час | % Соотвествия расчетанной интенсивности статистики |
| Главная Welcome страница | 520 | 547 | 5% |
| Вход в систему | 422 | 447 | 5% |
| Переход на страницу поиска билетов | 305 | 323 | 5% |
| Заполнение полей для поиска билета | 282 | 285 | 1% |
| Выбор рейса из найденных | 270 | 285 | 5% |
| Оплата билета | 175 | 174 | -1% |
| Просмотр квитанций | 280 | 273 | -3% |
| Отмена бронирования | 73 | 73 | 0% |
| Выход из системы | 326 | 336 | 3% |
| Перход на страницу регистрации | 97 | 101 | 4% |
| Заполнение полей регистарции | 97 | 101 | 4% |
| Переход на следуюущий эран после регистарции | 97 | 101 | 4% |
| **Итого** | 2944 | 3043 | 3% |

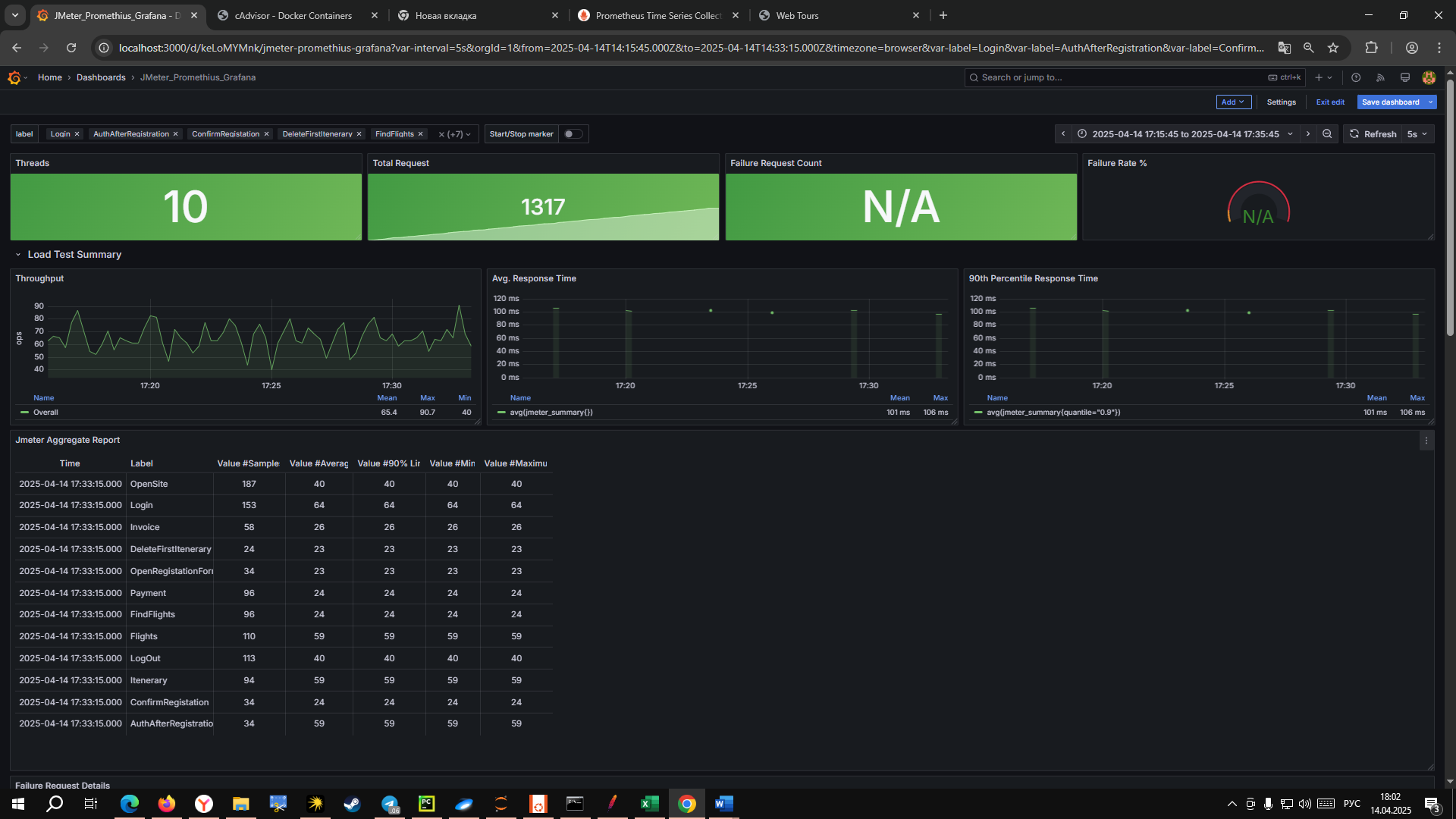


Рисунок – Итоге отладочного теста в графане

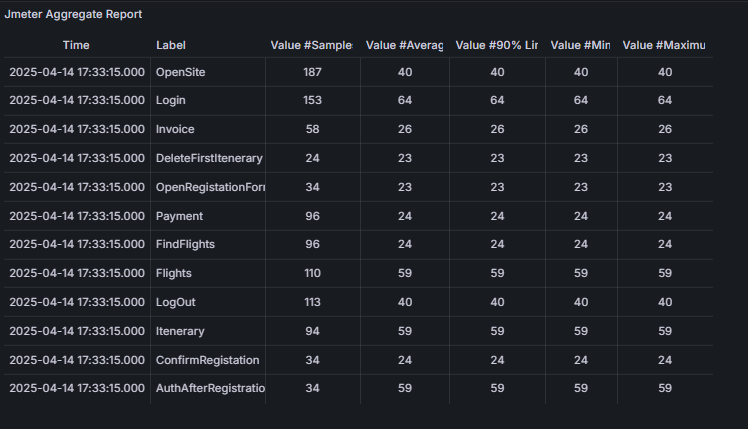


Рисунок – Табличка из графаны поближе

Таблица – Сопоставление нагрузки

|  |  |  |  |
| --- | --- | --- | --- |
| ScriptName | Расчетная интенсивность запросов / 20 мин | Фактическая интенсивность в тесте | % Отклонение от Профиля |
| OpenSite | 182 | 187 | 2% |
| Login | 149 | 153 | 3% |
| Flights | 108 | 110 | 2% |
| FindFlights | 95 | 98 | 3% |
| Payment | 95 | 96 | 1% |
| Invoice | 58 | 58 | 0% |
| Itenerary | 91 | 94 | 3% |
| DeleteFirstItenerary | 24 | 24 | -1% |
| Logout | 112 | 113 | 1% |
| OpenRegistationForm | 34 | 34 | 1% |
| ConfirmRegistation | 34 | 34 | 1% |
| AuthAfterRegistration | 34 | 34 | 1% |
|  | 1014 | 1035 | 2% |

## Критерии корректности проведенных тестов

|  |  |  |
| --- | --- | --- |
| Название критерия | Метрика | Максимальное значение |
| Время отклика транзакций | 90 percentile response time | 2,5 сек |
| Количество ошибок по транзакциям | Summary Report fail / pass + fail | 5% |
| Отклонение фактической нагрузки от профиля | Summary Report Errort % | - |
| Утилизация CPU контейнера WebTours | container\_cpu\_usage\_seconds\_total | 90% |
| Утилизация оперативной памяти контейнера WebTours | container\_memory\_rss | 90% |
| Утилизация диска WebTours | Disk IO | 90% |

# Наполнение БД

## Перенос данных с промышленной среды

### Требования к объемам данных

В системе зарегистрировано 50 карточек пользователей, у каждого пользователя куплено по 5 билетов.

## Методология генерации данных

### Требования к генерации данных

Уникальные логин, пароль, имена пользователей и их адреса. Для каждого пользователя куплены 5 случайных билетов.

### Способ генерации данных

Через API запрашиваются случайные данные для американских пользователей со сложными паролями.