Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

"Уфимский университет науки и технологий"

Отчет по лабораторной работе № 4  
«Нагрузочное тестирование веб-приложения. Разработка графического интерфейса клиентского приложения»

по дисциплине «Администрирование информационных систем»

Выполнили:

Студент группы МО-423

Чурмаев М. Р.  
Студент группы МО-424

Нуртдинов Д.М.

Проверила:

Сазонова Е. Ю.

Уфа 2025

Теоретическая часть:

1)**Нагрузочное тестирование** (load testing) — это тип тестирования производительности, целью которого является определение поведения системы при ожидаемой или увеличенной нагрузке. Оно помогает выявить, сколько пользователей, запросов или данных может обрабатывать система без ухудшения производительности.

Основные задачи нагрузочного тестирования:

* Проверить способность системы обрабатывать заданную нагрузку.
* Оценить производительность, стабильность и масштабируемость веб-приложения.
* Определить узкие места и сбои в системе.

Нагрузочное тестирование важно проводить до выпуска продукта, чтобы гарантировать его надежную работу в реальных условиях.

**Виды тестов производительности**

|  |  |
| --- | --- |
| **Вид тестирования** | **Описание** |
| **Нагрузочное тестирование (Load Testing)** | Проверка поведения приложения при определенной предполагаемой нагрузке. |
| **Стресс-тестирование (Stress Testing)** | Проверка поведения приложения при нагрузке, превышающей нормальную, чтобы выявить пределы устойчивости. |
| **Тестирование стабильности (Soak Testing)** | Долговременная нагрузка для выявления утечек памяти или снижения производительности с течением времени. |
| **Тестирование масштабируемости (Scalability Testing)** | Определение способности приложения масштабироваться при увеличении нагрузки (например, пользователей). |
| **Пиковое тестирование (Spike Testing)** | Внезапное увеличение нагрузки для проверки устойчивости приложения к резким всплескам. |

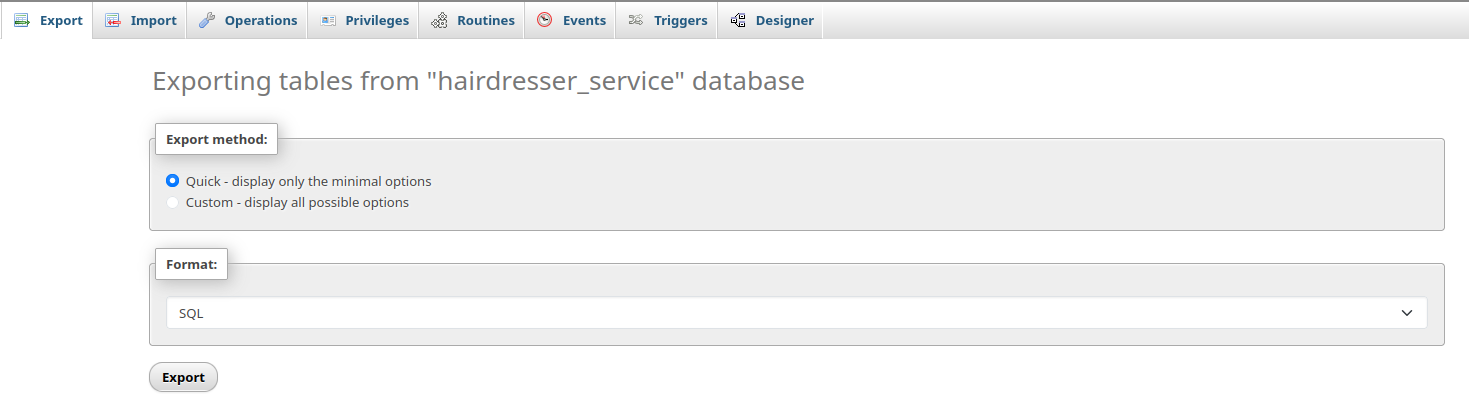
**Популярный инструментарий для нагрузочного тестирования**

|  |  |  |
| --- | --- | --- |
| **Инструмент** | **Описание** | **Особенности** |
| **Apache JMeter** | Открытый инструмент для тестирования производительности | Простота в использовании, поддержка разных протоколов. |
| **Locust** | Инструмент на Python для написания тестов в виде скриптов | Высокая гибкость, масштабируемость. |
| **Gatling** | Инструмент на Scala для тестирования HTTP-приложений | Подходит для сложных сценариев, хорошая визуализация. |
| **k6** | Мощный инструмент для нагрузочного тестирования на JavaScript | Интеграция с CI/CD, низкое потребление ресурсов. |
| **Artillery** | Легковесный инструмент для тестирования API и микросервисов | Отлично подходит для REST API тестов. |

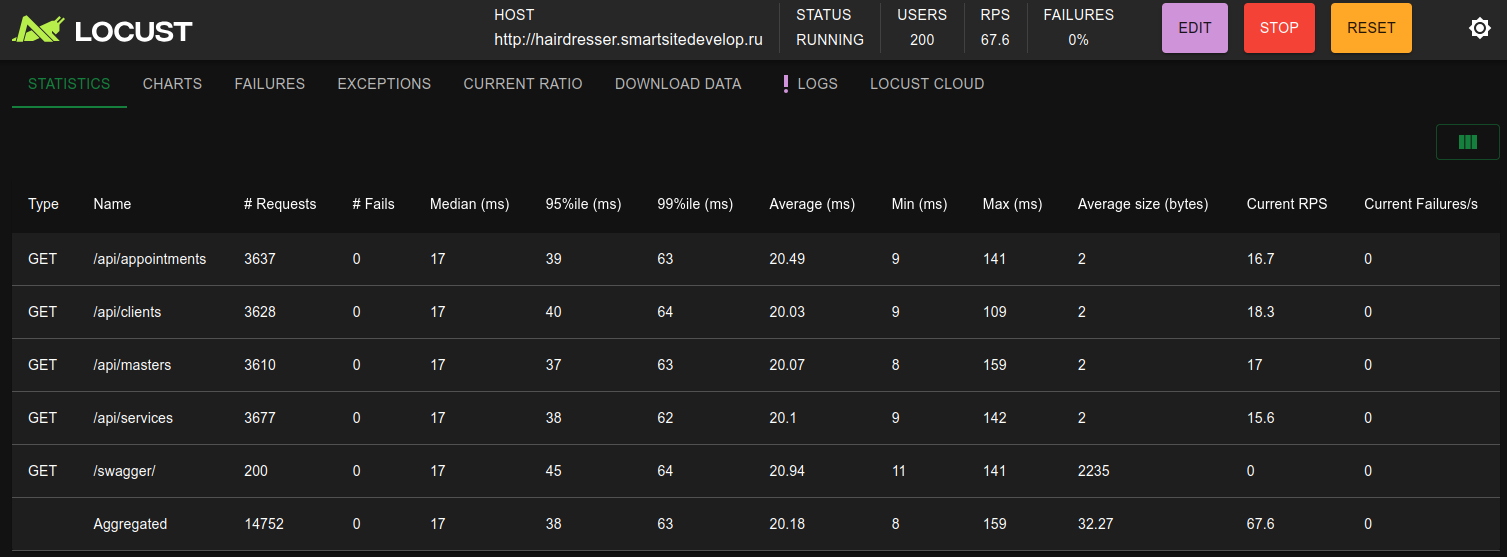
1. Проводим нагрузочное тестирование нашего веб-приложения, используя 3 подхода: нагрузочный, объемный и стрессовый.

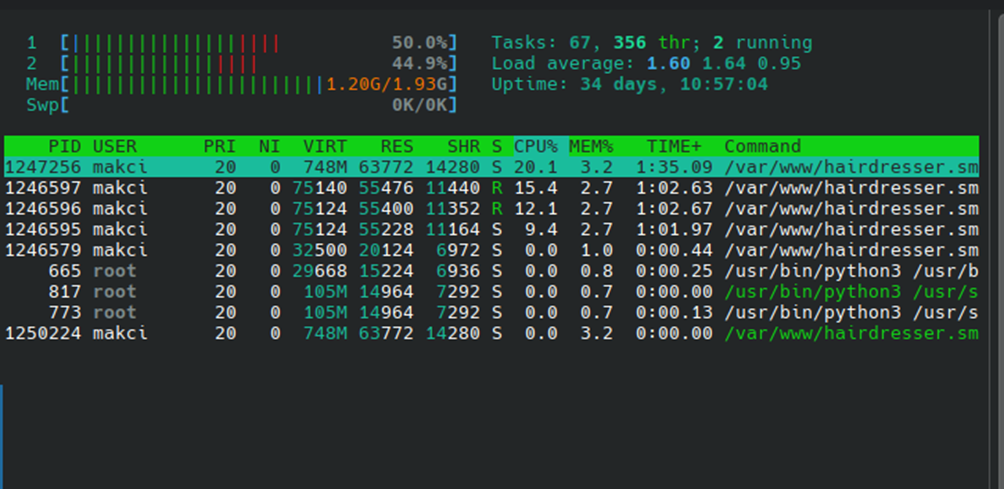
Ход выполнения работы:

1. Создаём бэкап базы данных

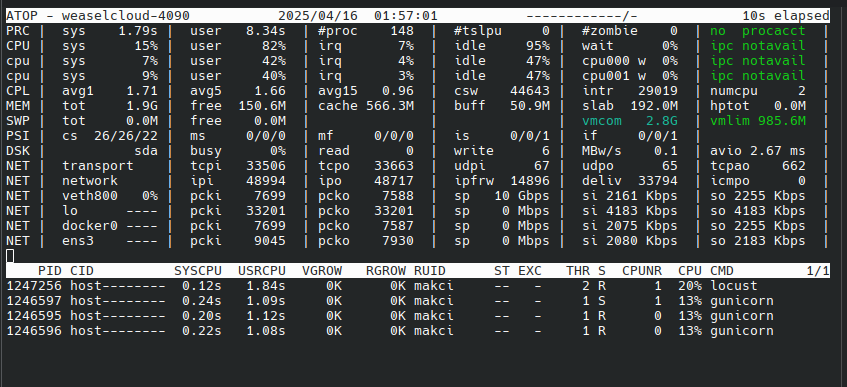


1. Проведём тест с GET запросами и количеством одновременных пользователей 200

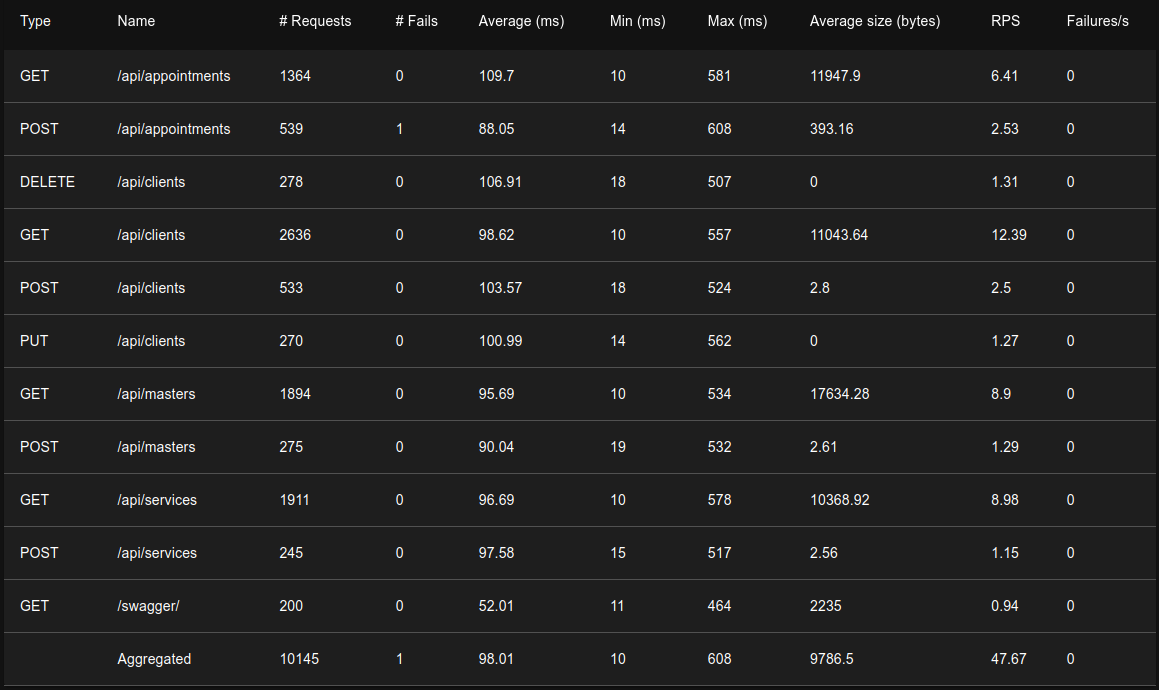


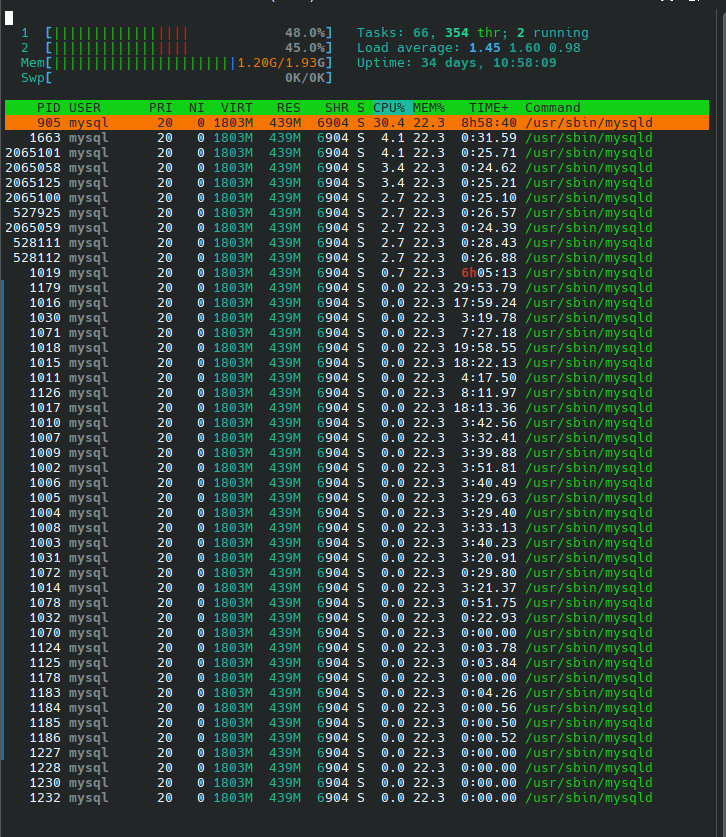
Скриншот с htop.  


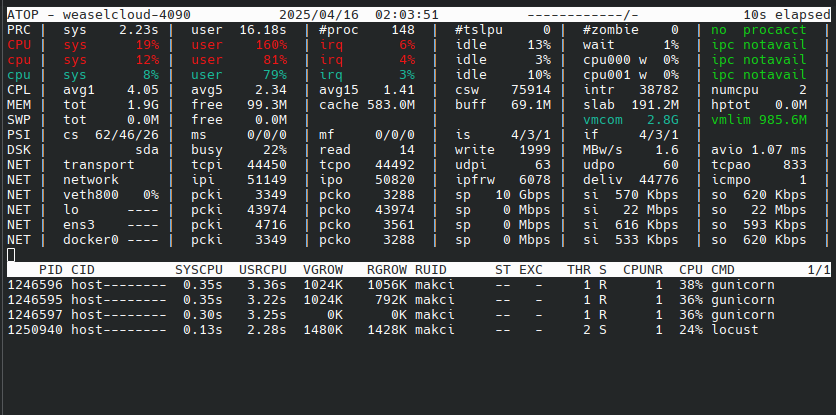
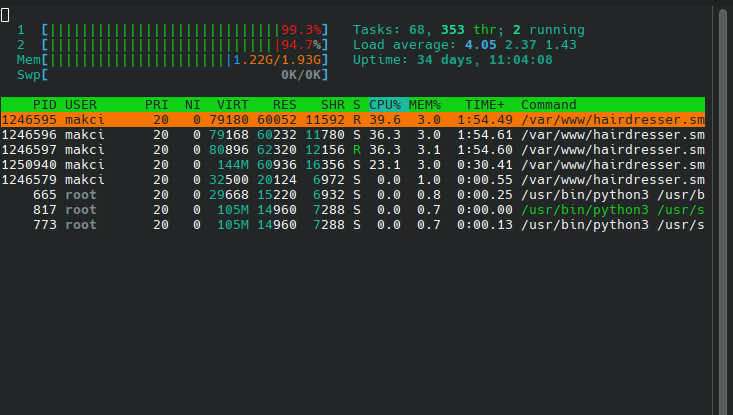
Скриншот с atop.



Включили в тестирование POST-запросы и зафиксировали результаты.

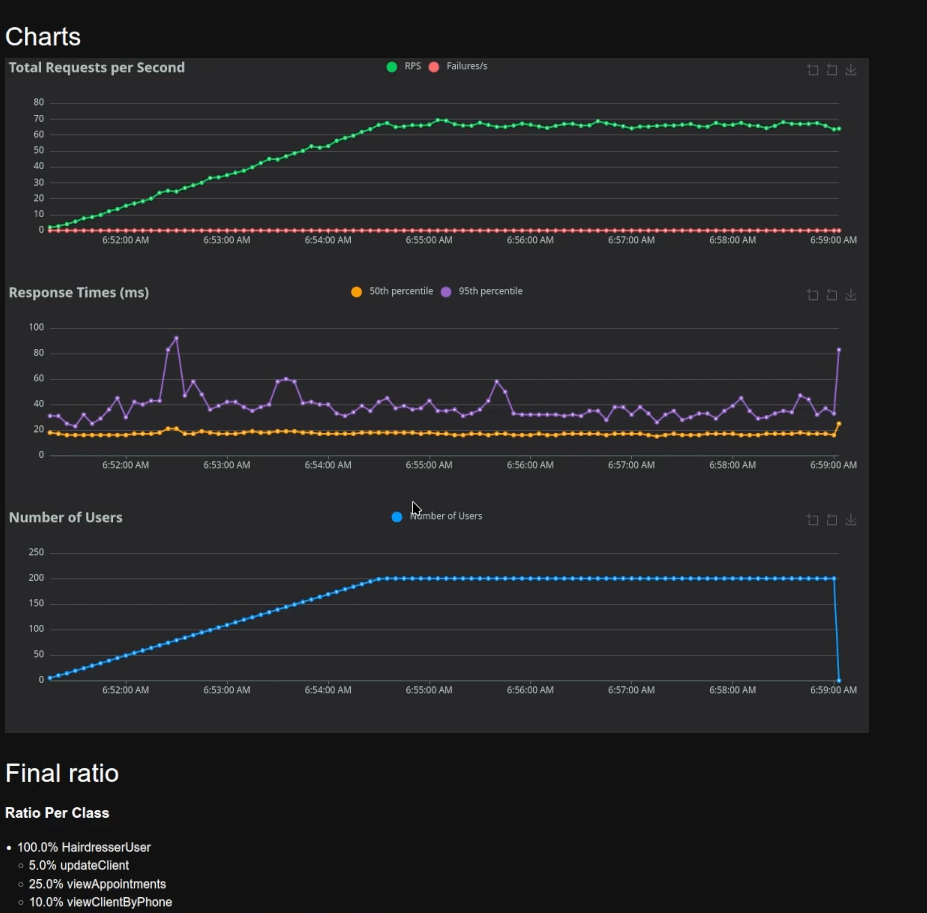




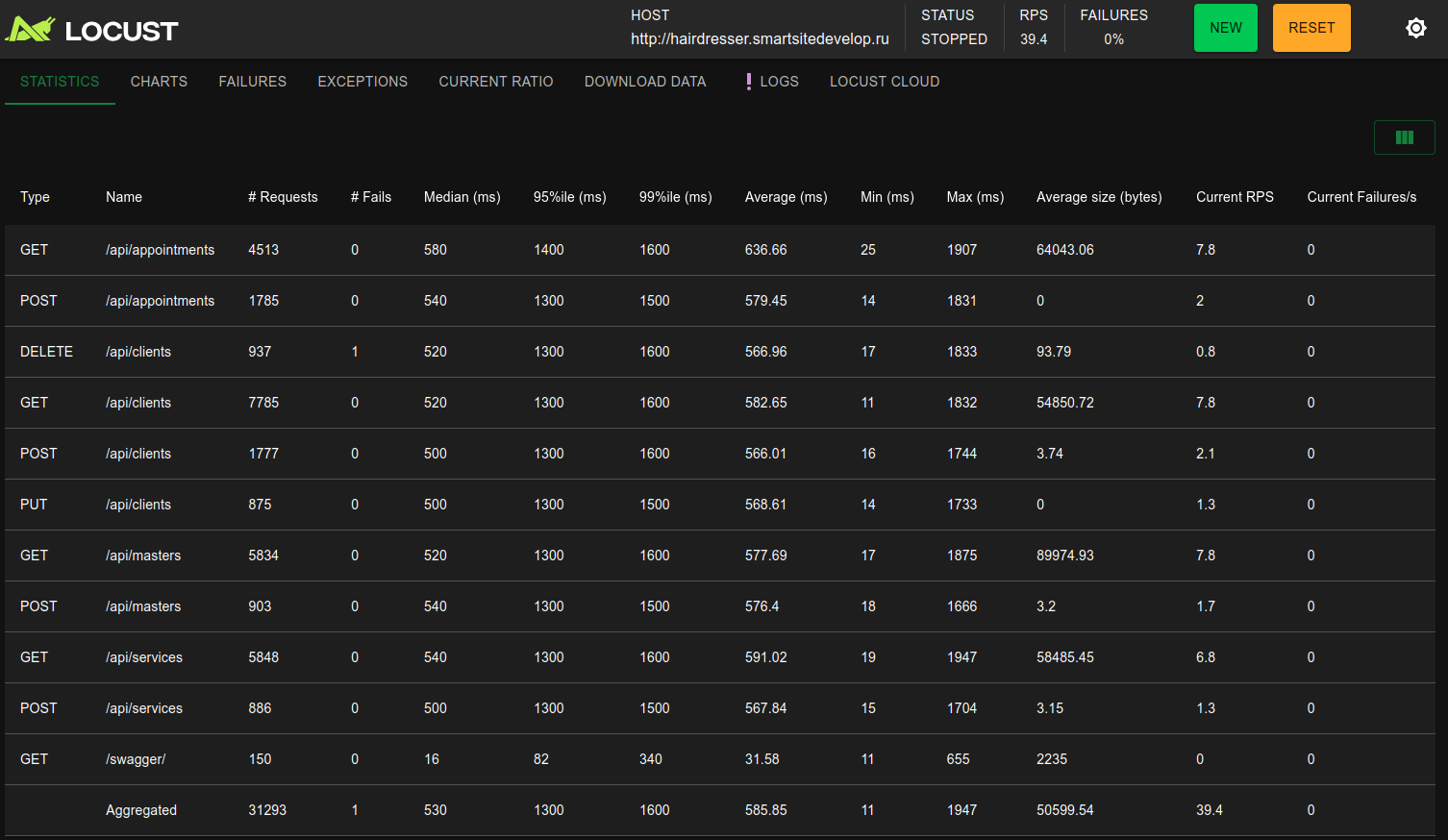


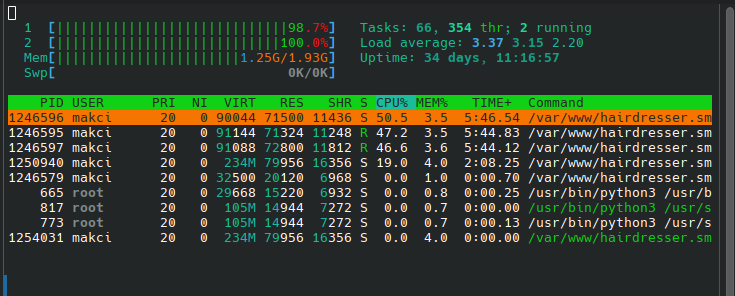
В ходе тестирования процессор работал почти на максимум.

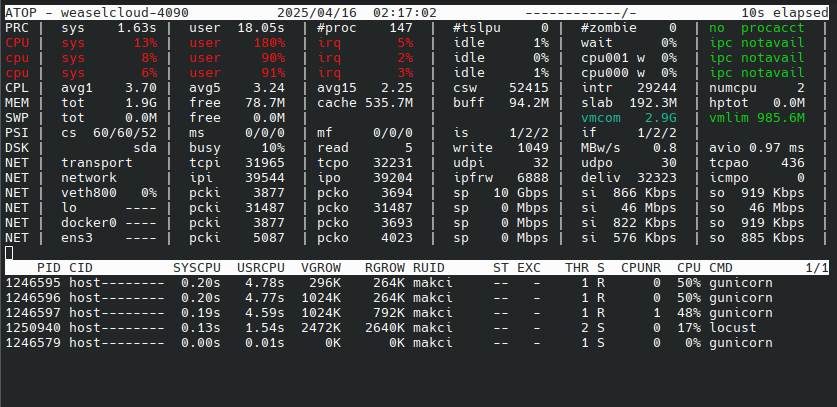
График



Подобрали максимальное значение эмулируемых пользователей таким образом, чтобы общая доля отказов (FAILURES) оставалась в пределах не более 1% , а нагрузка на CPU была в пределах 90-100% в течение 5 минут.







Процессор является наиболее нагруженным элементом сервера. Возможным вариантом решения может быть покупка более мощного процессора.