Федеральное государственное автономное образовательное учреждение высшего образования

 Национальный исследовательский университет ИТМО

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 Программная инженерия

Дисциплина «*Тестирование программного обеспечения*»

**Отчёт**

**по лабораторной работе №4**

Студент:

*Барсуков Максим Андреевич*

группа *P3315*

Преподаватель:

*Цопа Евгений Алексеевич*

г. Санкт-Петербург, 2025 г.

**Текст задания**

С помощью программного пакета [Apache JMeter](http://jmeter.apache.org/) провести нагрузочное и стресс-тестирование веб-приложения в соответствии с вариантом задания.

В ходе нагрузочного тестирования необходимо протестировать 3 конфигурации аппаратного обеспечения и выбрать среди них наиболее дешёвую, удовлетворяющую требованиям по максимальному времени отклика приложения при заданной нагрузке (в соответствии с вариантом).

В ходе стресс-тестирования необходимо определить, при какой нагрузке выбранная на предыдущем шаге конфигурация перестаёт удовлетворять требованиями по максимальному времени отклика. Для этого необходимо построить график зависимости времени отклика приложения от нагрузки.

**Параметры тестируемого веб-приложения:**

* URL первой конфигурации ($ 3100) - [http://stload.se.ifmo.ru:8080?token=495351464&user=-2105800574&config=1](http://stload.se.ifmo.ru:8080/?token=495351464&user=-2105800574&config=1);
* URL второй конфигурации ($ 4500) - [http://stload.se.ifmo.ru:8080?token=495351464&user=-2105800574&config=2](http://stload.se.ifmo.ru:8080/?token=495351464&user=-2105800574&config=2);
* URL третьей конфигурации ($ 8300) - [http://stload.se.ifmo.ru:8080?token=495351464&user=-2105800574&config=3](http://stload.se.ifmo.ru:8080/?token=495351464&user=-2105800574&config=3);
* Максимальное количество параллельных пользователей - 12;
* Средняя нагрузка, формируемая одним пользователем - 20 запр. в мин.;
* Максимально допустимое время обработки запроса - 520 мс.

**Выполнение**

**Исходный код**

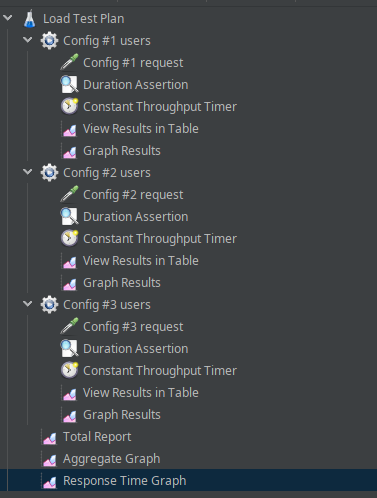


<https://github.com/maxbarsukov/itmo/tree/master/6%20%D1%82%D0%BF%D0%BE/%D0%BB%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D1%8B%D0%B5/lab4>

**Описание конфигурации JMeter для нагрузочного тестирования**

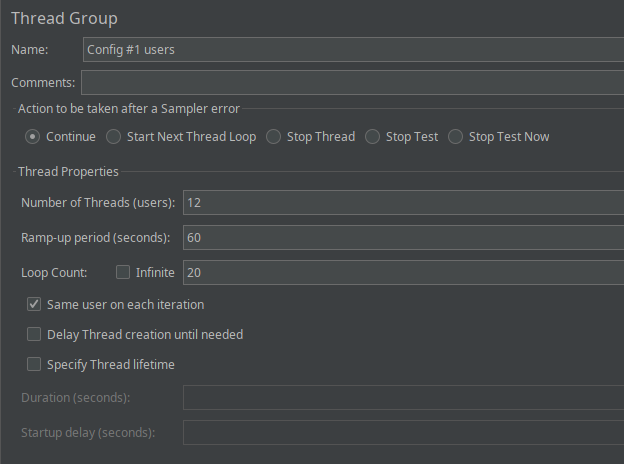
**Общая конфигурация**

Конфигурация нагрузочного тестирования состоит из 3 Thread Group объектов, по 1 на каждую конфигурацию, конфигурация каждого из них идентична, за исключением параметра config при формировании запроса. Помимо объектов Thread Group, присутствуют вспомогательные объекты для генерации общих отчетов.



**Конфигурация Thread Group**

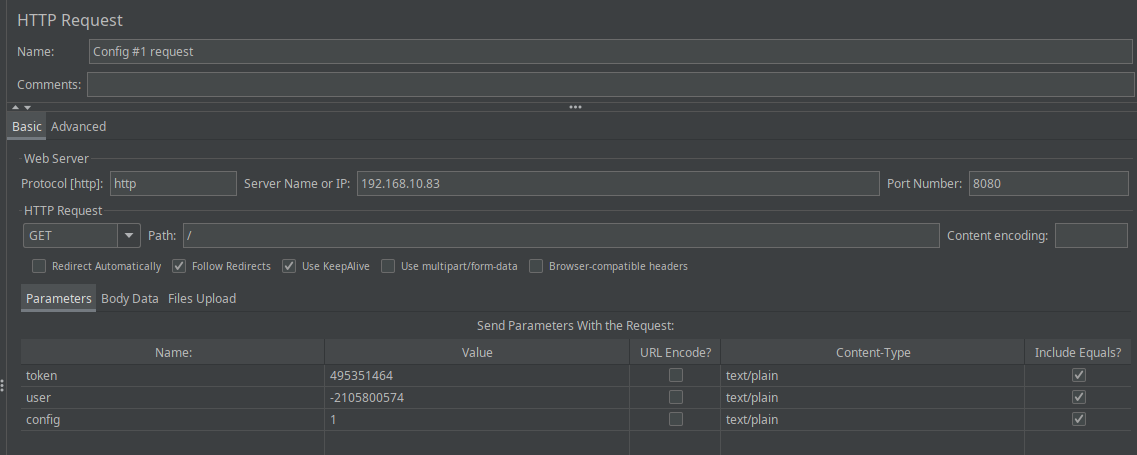
В конфигурации Thread Group используются настройки количества пользователей (Number of Threads) и количество запусков теста (Loop Count).



Также thread group содержит следующие под-объекты:

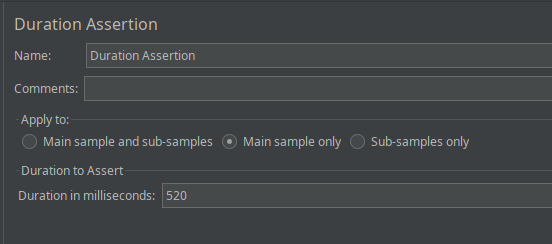
**HTTP Request**

осуществляет запросы к тестовому серверу



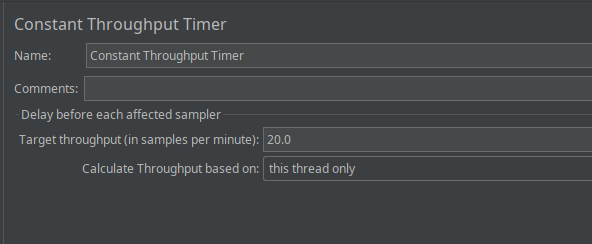
**Duration Assertion**

осуществляет проверку времени отклика сервера



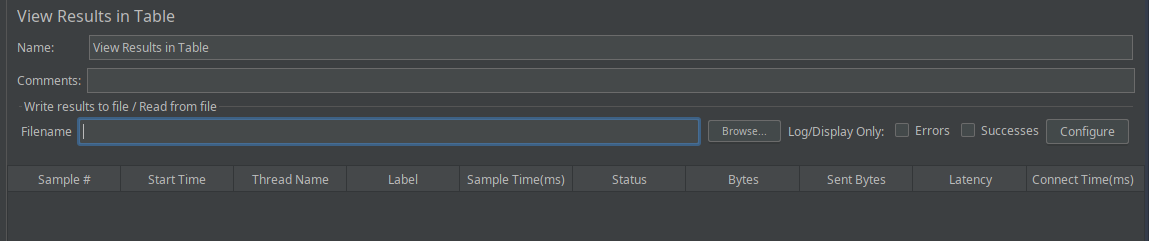
**Constant Throughput Timer**

определяет среднюю нагрузку, формируемую одним пользователем в мин



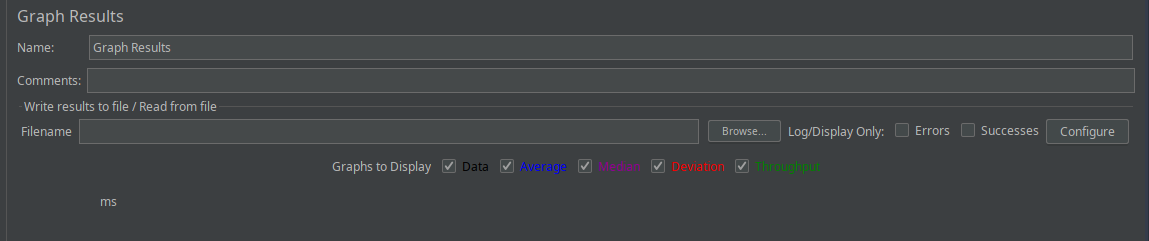
**View Results in Table**

Отображение результатов тестирования в виде таблицы



**Graph Results**

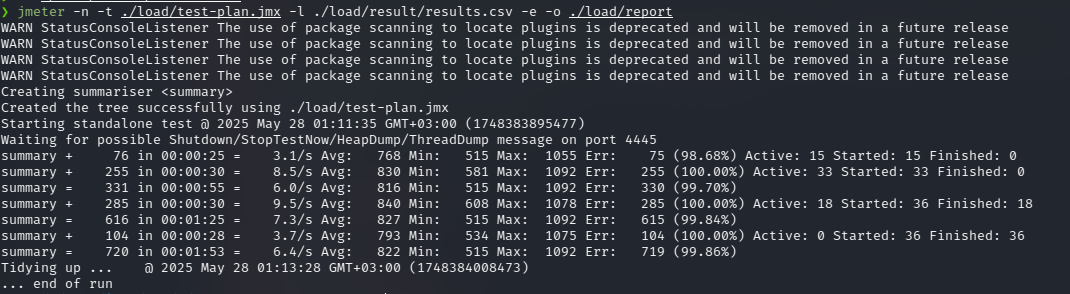
Отображение результатов тестирования в виде графика



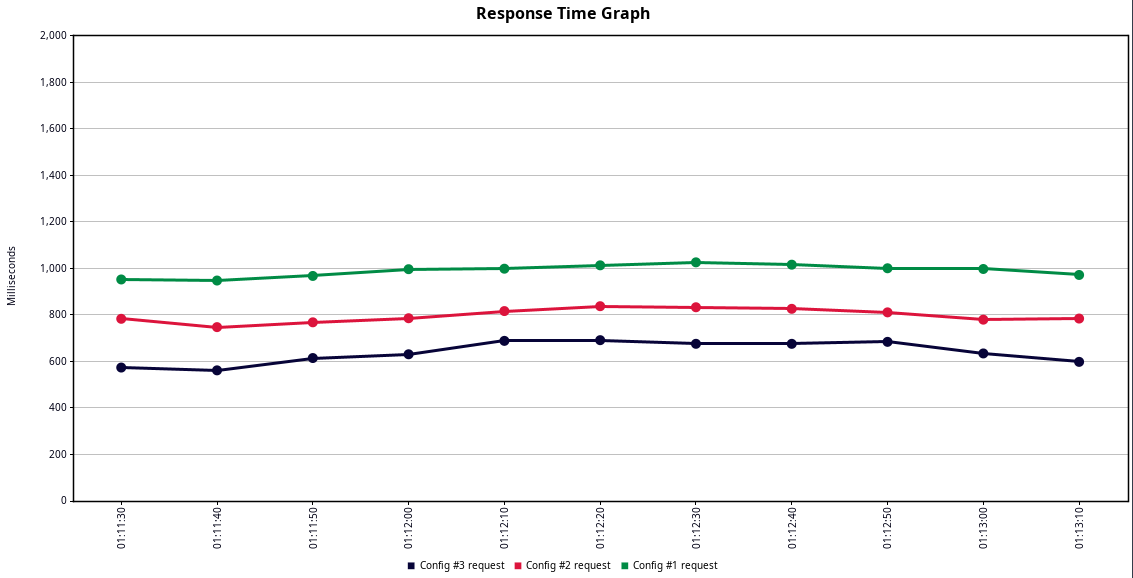
**Запуск сценария**

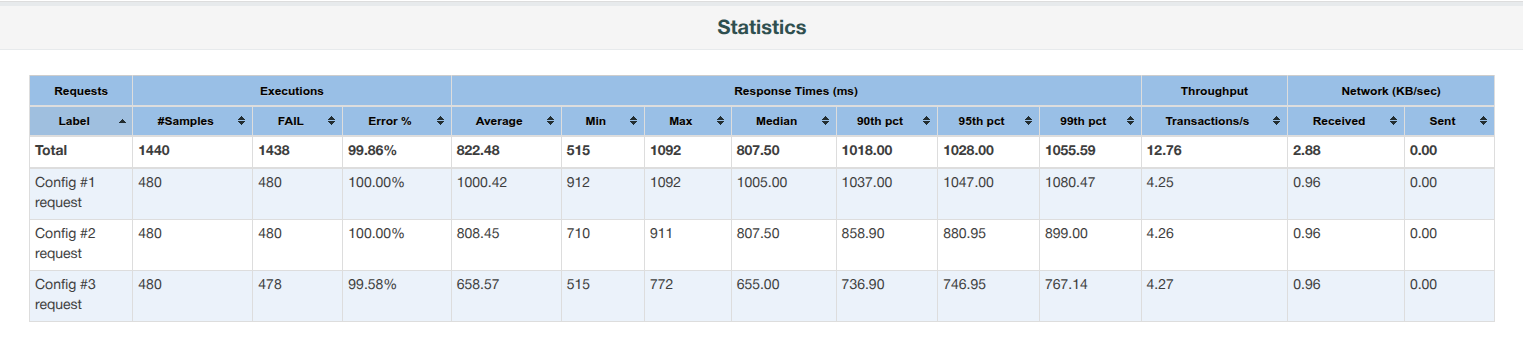
jmeter -n -t ./load/test-plan.jmx -l ./load/result/results.csv -e -o ./load/report

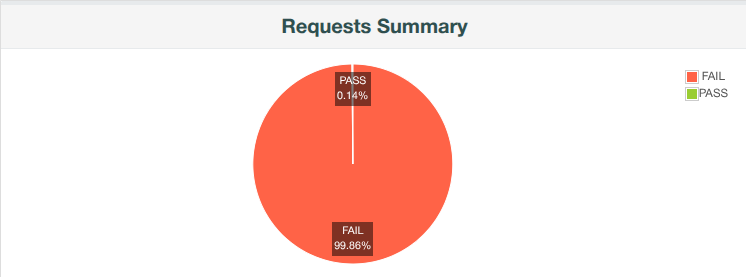
Результат:

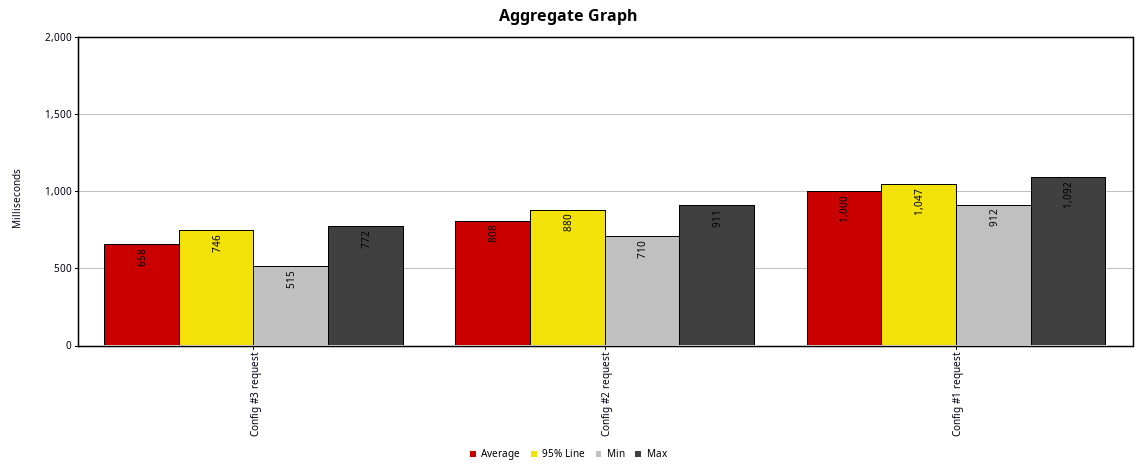


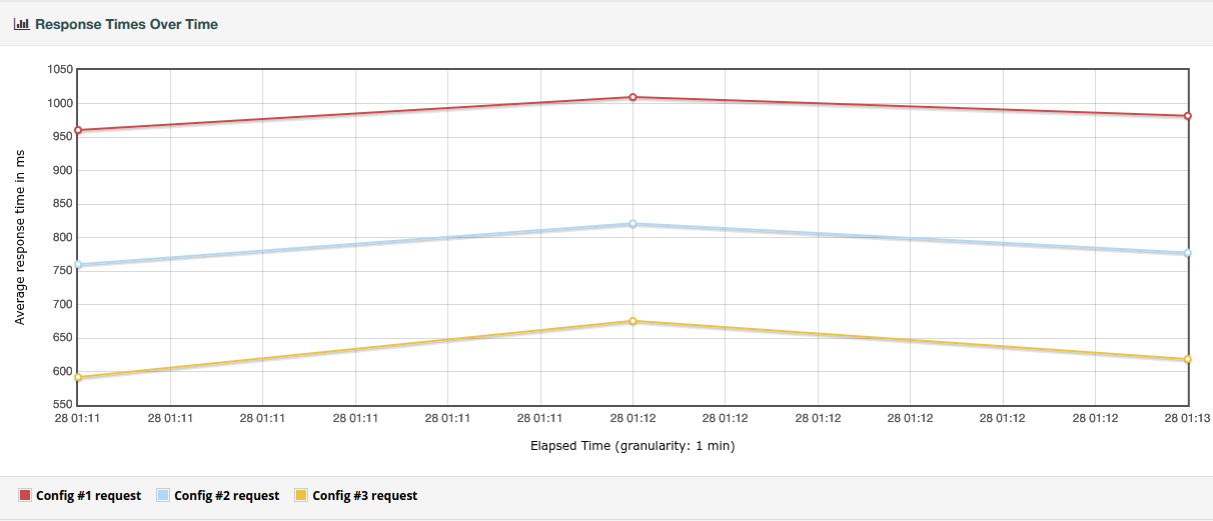
**Графики пропускной способности приложения, полученные в ходе нагрузочного тестирования**

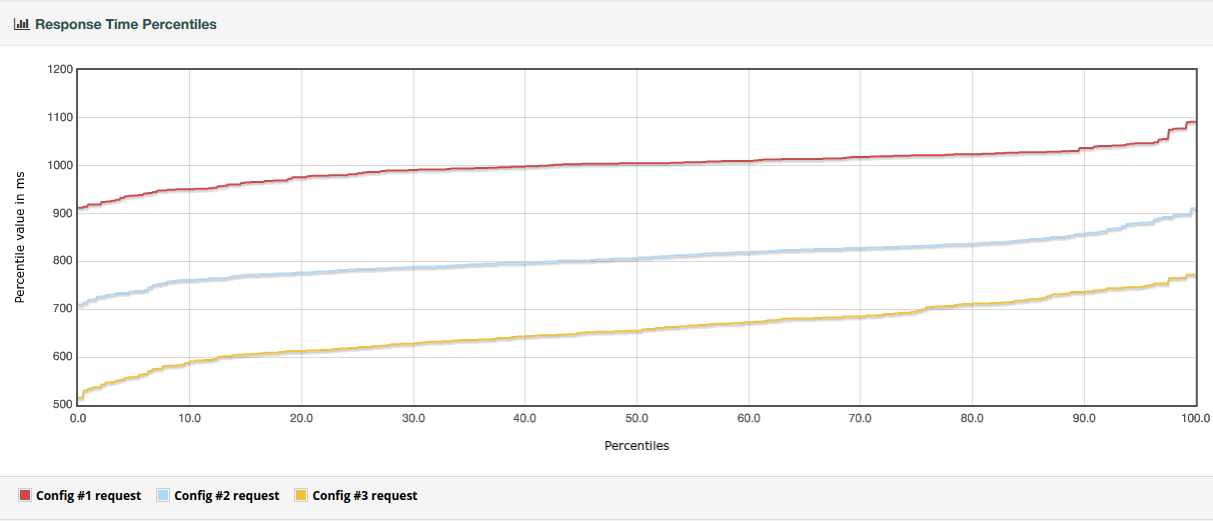


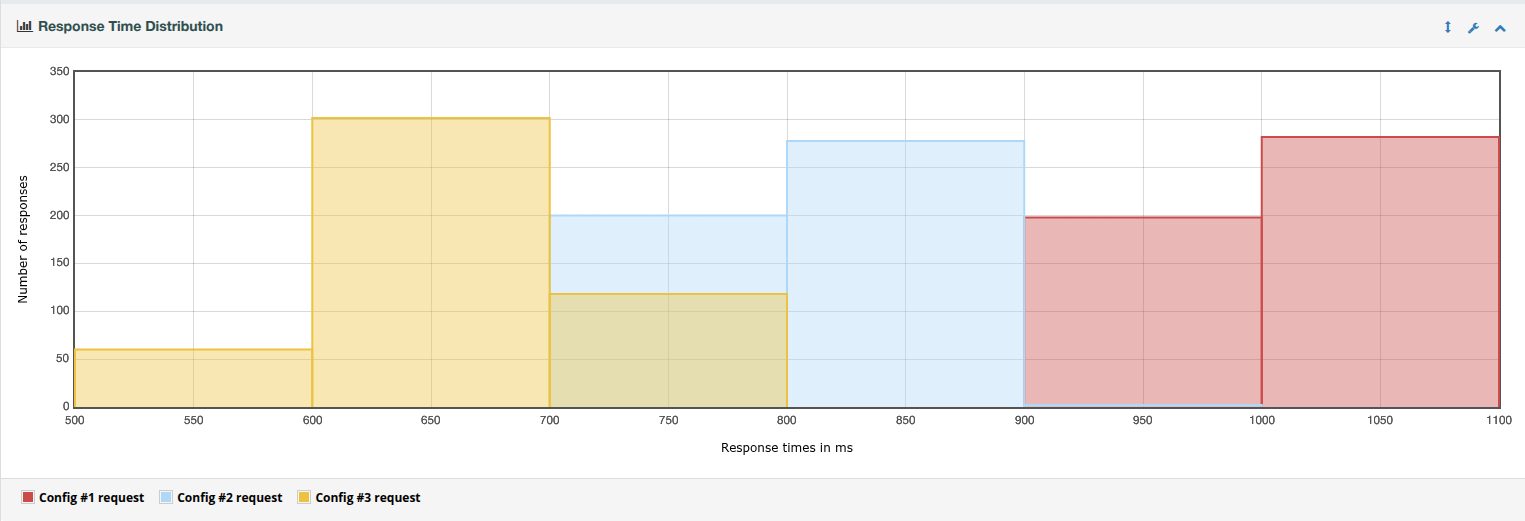




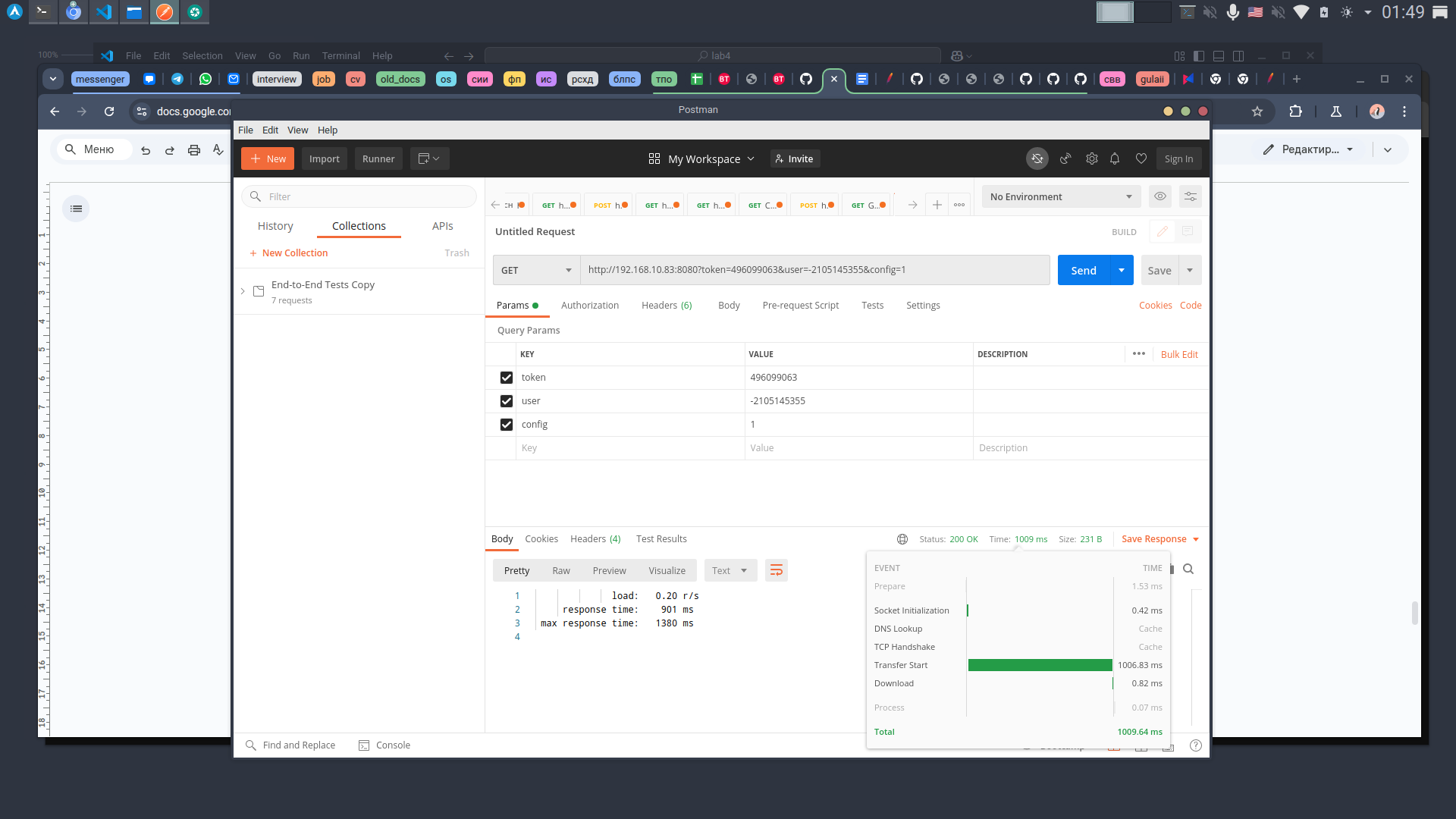


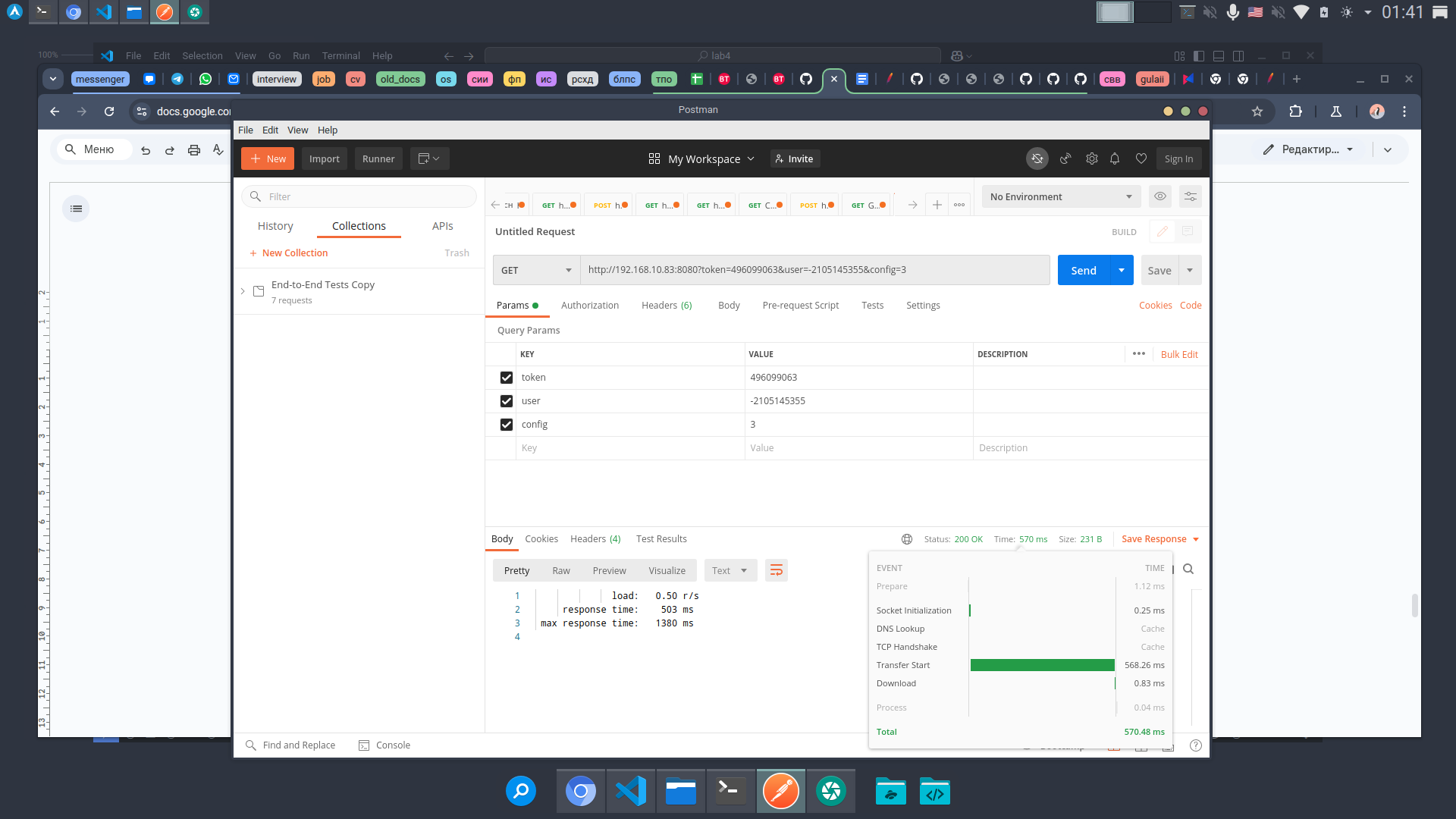






**Выводы по выбранной конфигурации аппаратного обеспечения**

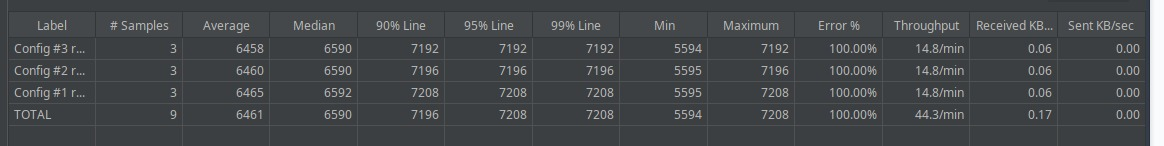
Для нагрузочного тестирования обратимся к графикам времени отклика запроса к количеству успешных откликов. На них можно увидеть, что в заданные вариантом временные рамки 520 ms укладываются в общем виде никакие конфигурации не укладываются в требования. Ближе всех в заданным рамкам конфигурация №3 – в то время как ее время обработки запроса ≥500 мс, для конфигурации №2 это ≥700 мс, а для №1 и вовсе ≥ 900 мс.



По этой же причине, общий процент неудачных запросов во время тестирования крайне высок. Для среднего и медианного времени обработки запроса конфигурация №3 также лучше конфигурации №2 на 200 мс и на 400 мс лучше конфигурации №1. Исходя из этого нам следует выбрать конфигурацию №3, хотя и она в >95% не вписывается в рамки 520 мс. В общем же случае все три конфигурации не соответствуют требованиям. Другими словами…



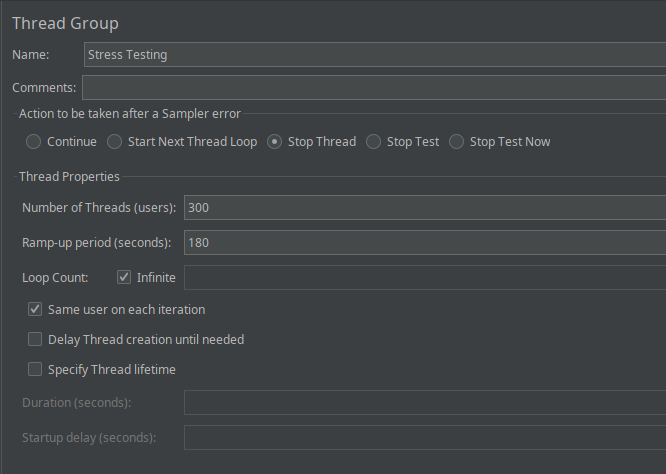
Однако стоит отметить, что настолько неудовлетворительная ситуация для 3-го, самого дорогого сервера вызвана задержками в сети между внутренней сетью кафедры и среды, в которой производится тестирование. Например, случались такие перлы:

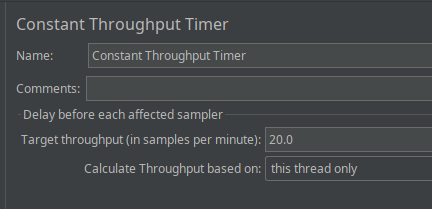
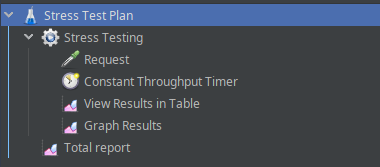


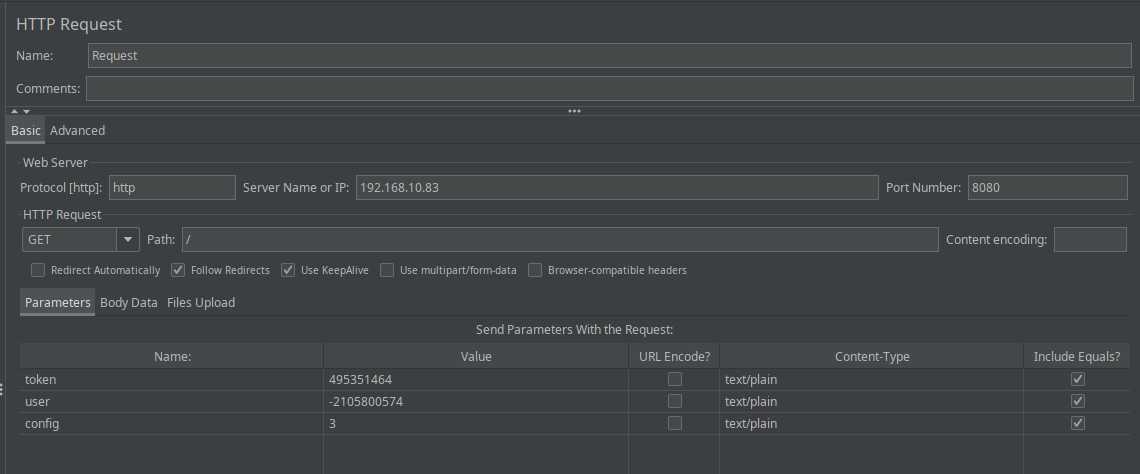
Но для конечный пользователь вряд ли будет находиться во внутренней сети, а значит результаты тестирования можно считать действительными.

**Описание конфигурации JMeter для стресс-тестирования**

Для стресс-тестирования была выбрана третья конфигурация оборудования как единственная, минимально подходящая под требования и имеет лучшие шансы показать достойный результат. Из нагрузочного тестирования мы знаем, что при 12 пользователях сервис уже не удовлетворяет требованиями по максимальному времени отклика, поэтому Duration Assertion был убран – мы будем пытаться положить сервер. Конфигурация настроена таким образом, чтобы постепенно увеличивать количество пользователей, при этом считая что каждый пользователь отправляет в среднем 20 запр. в мин до тех пор, пока сервер не начнет возвращать ошибку 503 – Service unavailable.

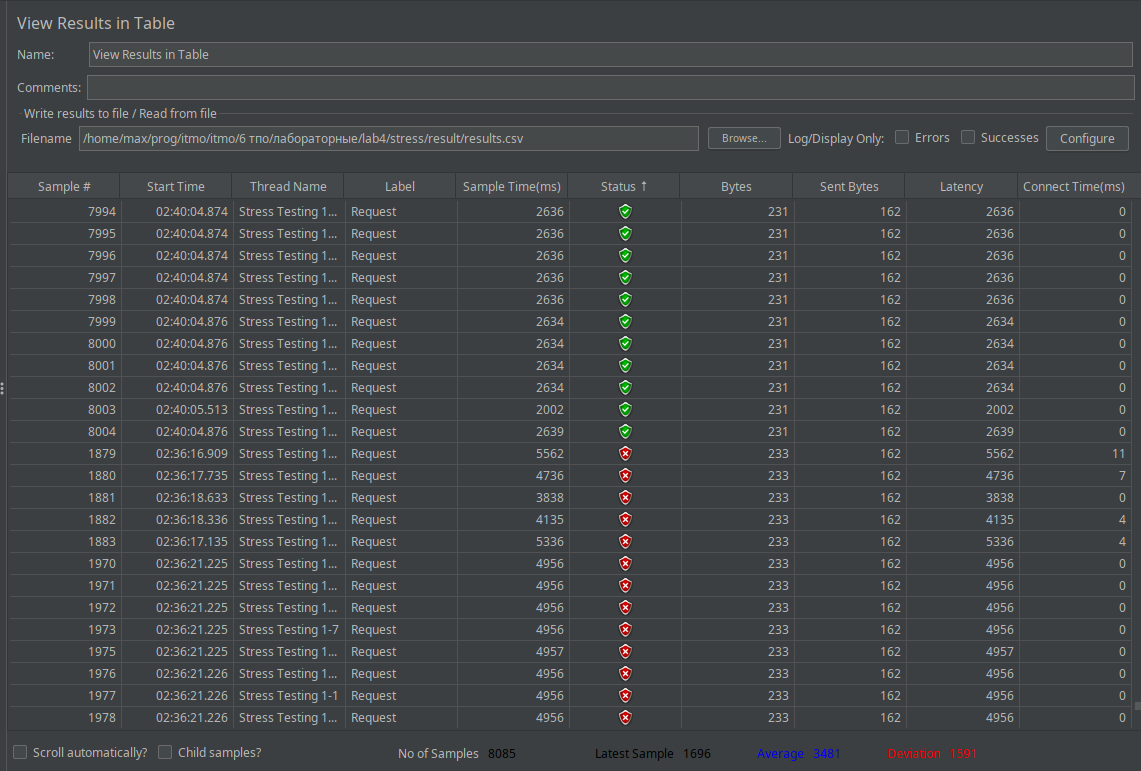
**Общая конфигурация**

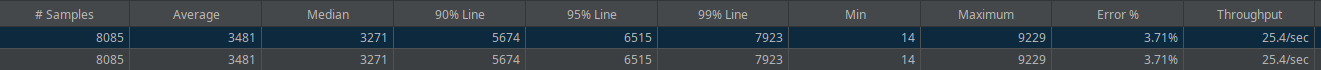


****

**График изменения времени отклика от нагрузки для выбранной конфигурации, полученный в ходе стресс-тестирования системы**

Получено 8085 записей JMeter об обращении к веб-приложению, из них 3.71% завершились ошибкой, макс. время отклика составило 9229 мс.



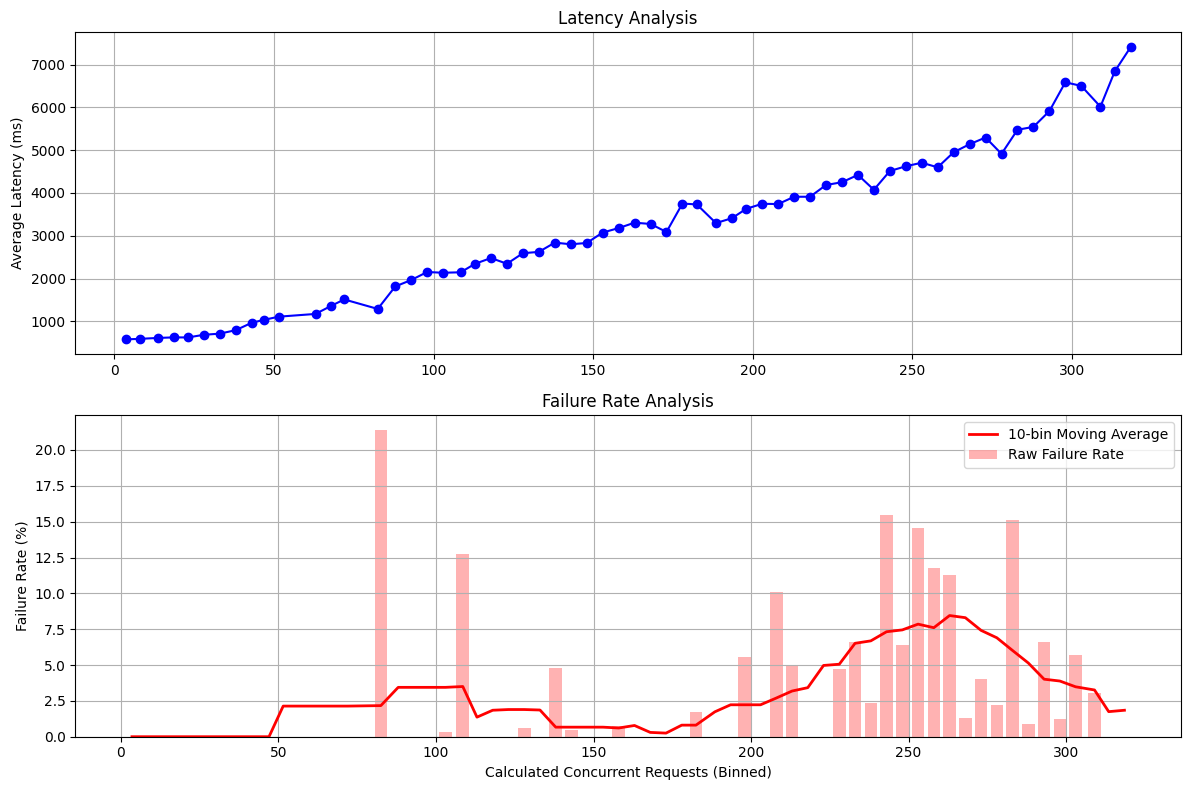


Для анализа результатов стресс тестирования, было написано несколько скриптов, читающий результаты из .csv файла, так как стандартных инструментов анализа Apache JMeter недостаточно.

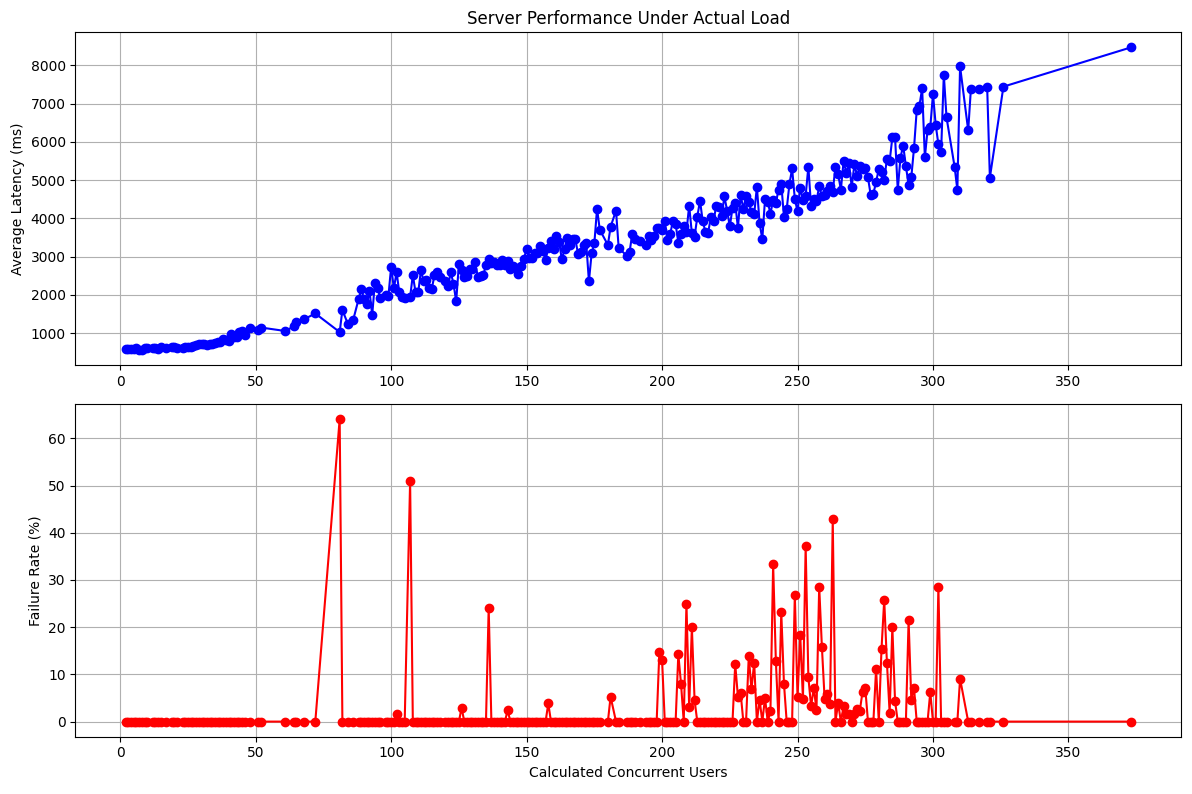
Попробуем проанализировать полученные результаты:

First failures observed at 81 concurrent users

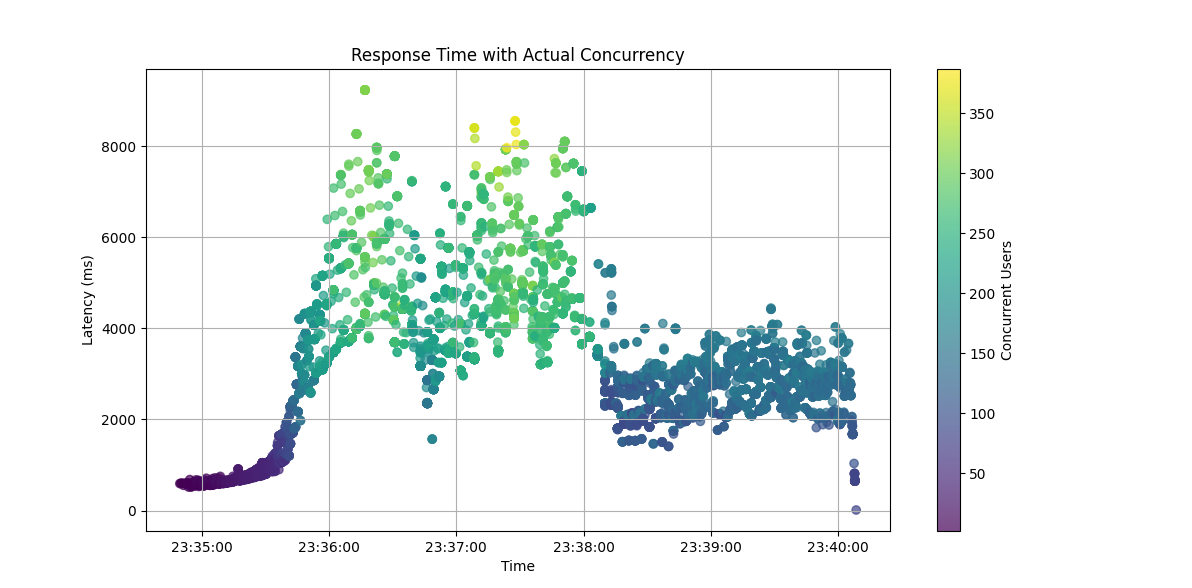
**Оценка задержки и частоты ошибок**

****

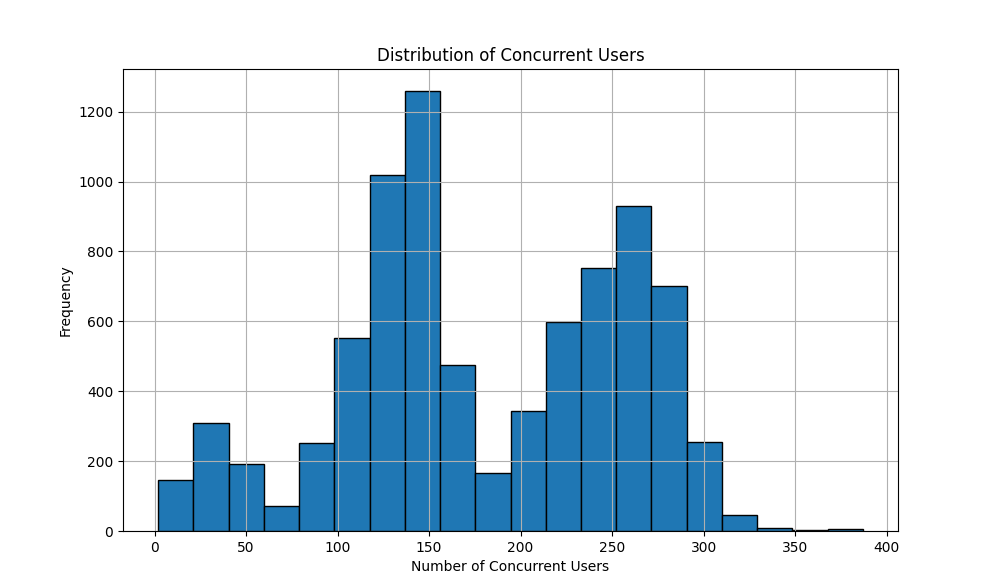
**Оценка средней задержки для кол-ва параллельных пользователей**

****

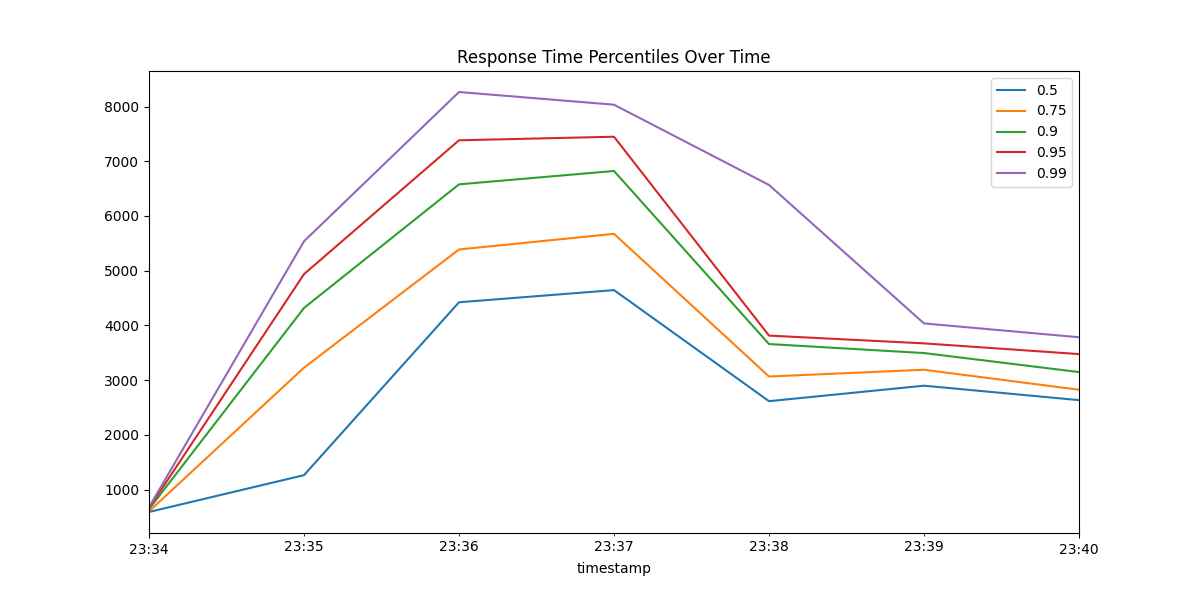
**Оценка времени отклика во времени от кол-ва параллельных пользователей**

****

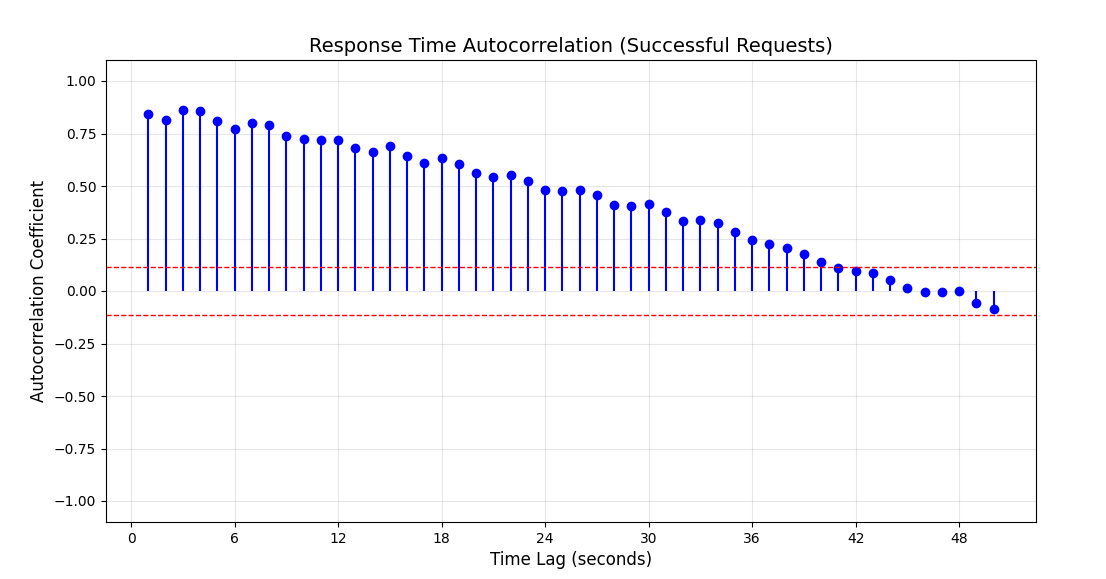
**Распределение параллельных пользователей**

****

**Перцентили времени обработки запроса во времени**

****

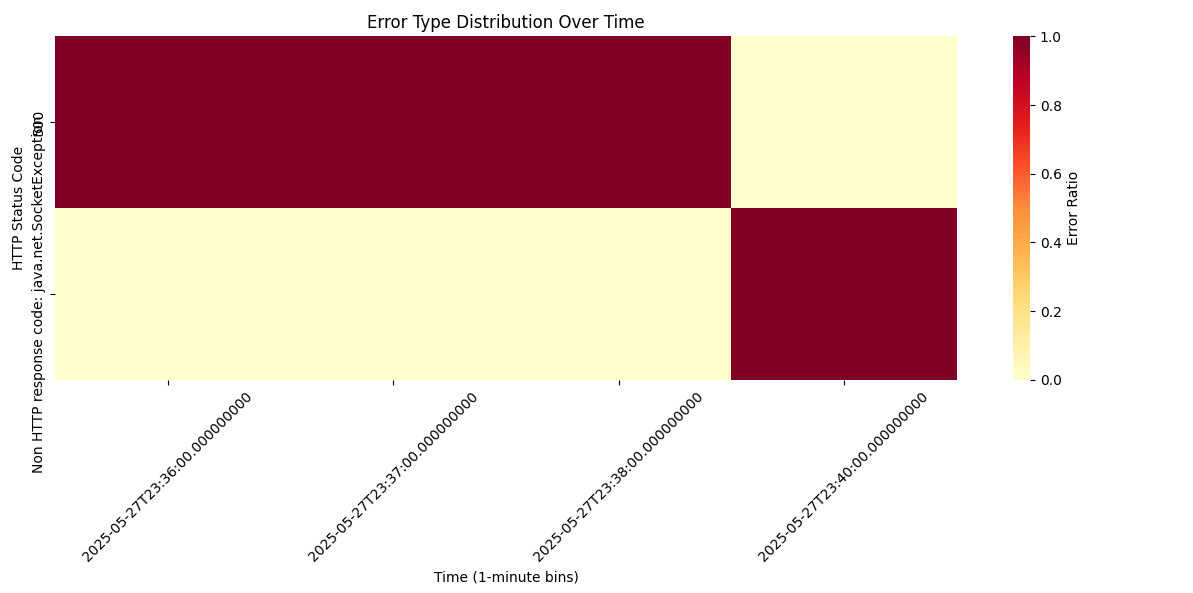
**График автокорреляции для времени отклика**

****

* Значимые корреляции (за пределами красных линий) указывают на периодические паттерны
* Положительная корреляция: высокие значения времени отклика имеют тенденцию следовать друг за другом

При сравнении с графиком количества запросов взаимосвязей не выявлено.

**Динамика распределения ошибок по типам HTTP-статусов**

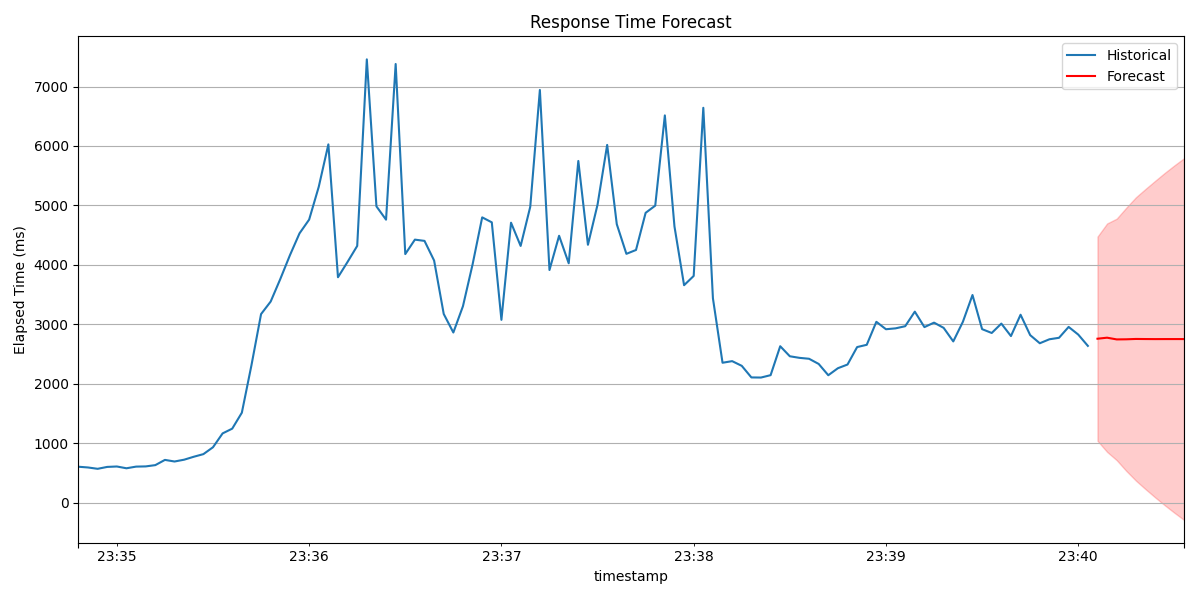


**Кластеризация ошибок во временном пространстве**



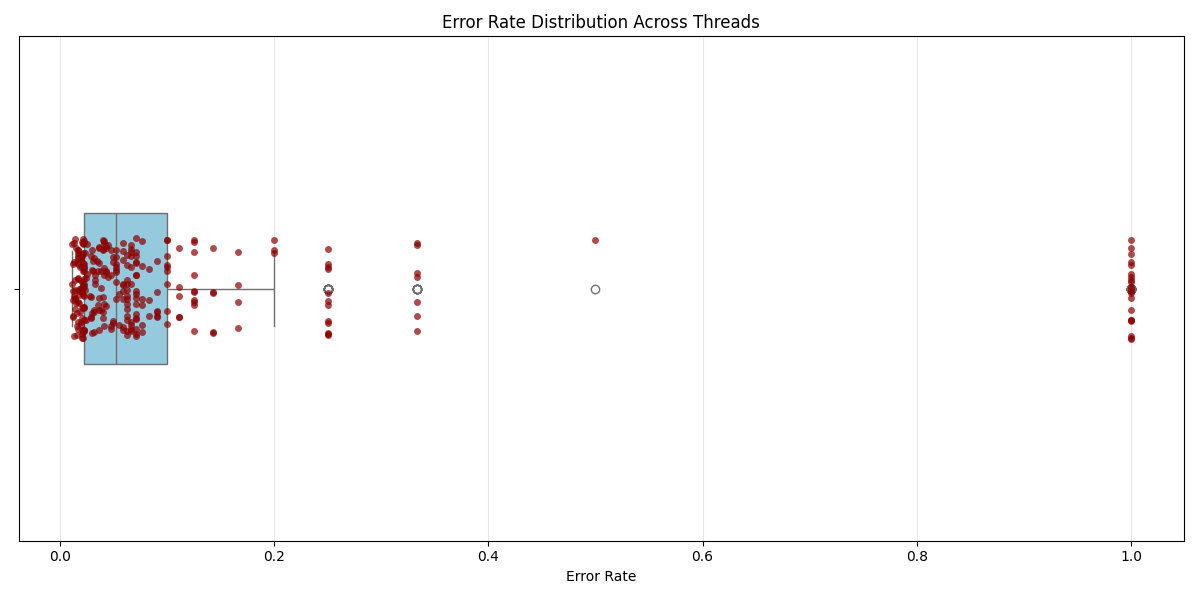
* Крупные кластеры (много точек одного цвета) → периоды нестабильности системы.
* Интервалы между кластерами → периоды стабильной работы
* Кластеры ошибок совпадают ли с пиками нагрузки.

**Прогнозирование деградации времени отклика**

****

Расширяющийся доверительный интервал → рост неопределенности в прогнозе. Нет явного роста прогнозируемых значений → нет явных признаков надвигающейся деградации.

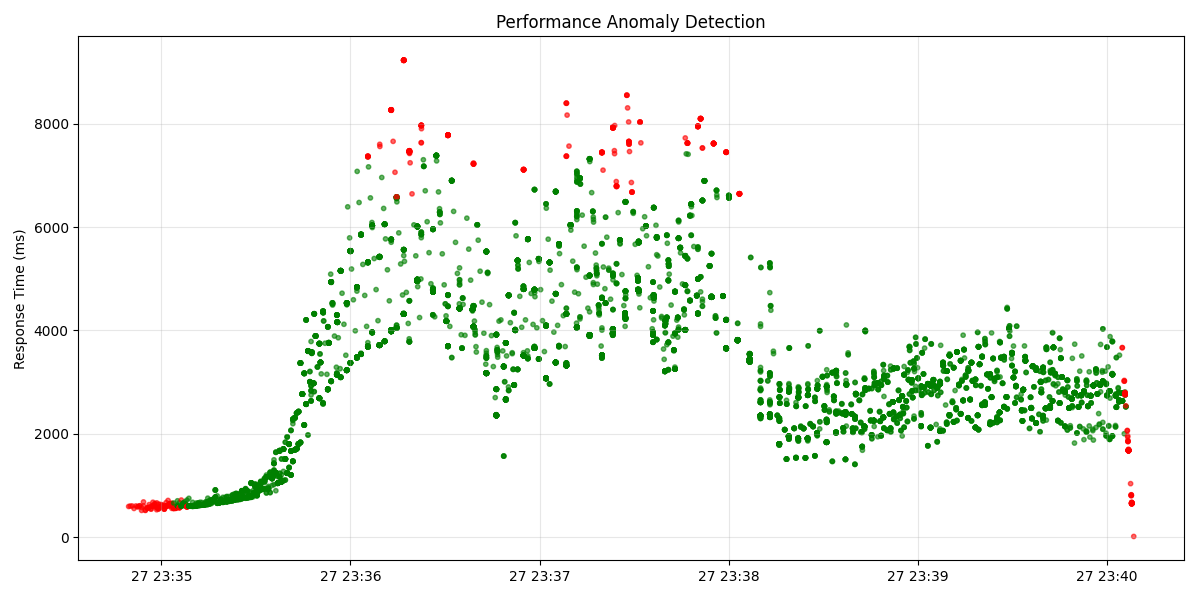
**Распределение частоты ошибок между потоками**

****

Выбросы (красные точки далеко от "ящика") → проблемные потоки

Ширина "ящика" → вариабельность ошибок между потоками. Узкий ящик → стабильная работа большинства потоков

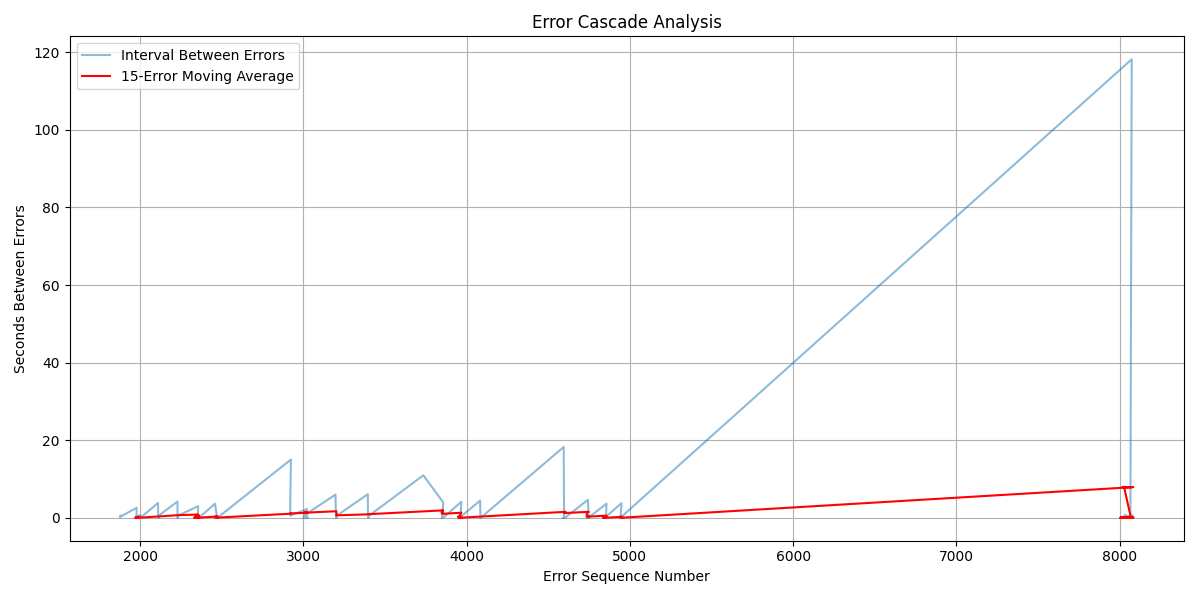
**Выявление аномалий в производительности**

****

Красные точки → аномальные запросы:

* Группы аномалий → системные проблемы (например, сбои БД)
* Одиночные аномалии → возможные сетевые артефакты или ошибки тестирования

**Анализ временных интервалов между ошибками**



Нисходящий тренд → ошибки учащаются (эффект домино).

Пики на графике → периоды восстановления после сбоев.

Среднее значение интервалов:

* < 1 сек → критическая нестабильность
* 1-5 сек → требуется мониторинг
* 5 сек → штатная работа

**Выводы по стресс-тестированию**

Система демонстрирует критическую неустойчивость даже к минимальным нагрузкам. Основные проблемы носят системный характер.

**Деградация производительности:**

Время отклика растет со скоростью +82 мс/интервал (30 сек). 5% запросов обрабатываются дольше 6.5 секунд. Среднее время установки соединения: 1.84 секунды.

**Возникающие ошибки:**

* 37% ошибок — HTTP 500 (внутренние сбои сервера).
* 27% ошибок — SocketException (разрывы соединений).

**Паттерны:**

* Кластеры ошибок каждые 0.76 секунд.
* Эффект домино: интервалы между ошибками сокращаются в 5 раз.

Выявлено 405 аномальных запросов (5% от общего числа). Характеристики аномалий: время отклика >7.9 секунд (99-й перцентиль), сочетание высокой латентности и нагрузки.

**Выводы**

В ходе выполнения лабораторной работы было проведено нагрузочное и стресс-тестирование веб-приложения с целью выявления самой дешевой конфигурации, удовлетворяющей требованиям. JMeter - удобный и гибкий инструмент для проведения нагрузочного тестирования. В результате выполнения лабораторной работы была выбрана 3 конфигурация, хоть и не удовлетворяющая требованиям, но имеющую лучшие шансы показать приемлемый результат. В ходе стресс-тестирования было определено, при какой нагрузке выбранная на предыдущем шаге конфигурация – перестает удовлетворять требованиями по максимальному времени отклика. Для этого был построен график зависимости времени отклика приложения от нагрузки.