

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](#) 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;>
- URL второй конфигурации (\$ 4500) - <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;>
- Максимальное количество параллельных пользователей - 12;
- Средняя нагрузка, формируемая одним пользователем - 20 запр. в мин.;
- Максимально допустимое время обработки запроса - 520 мс.

Выполнение

Исходный код

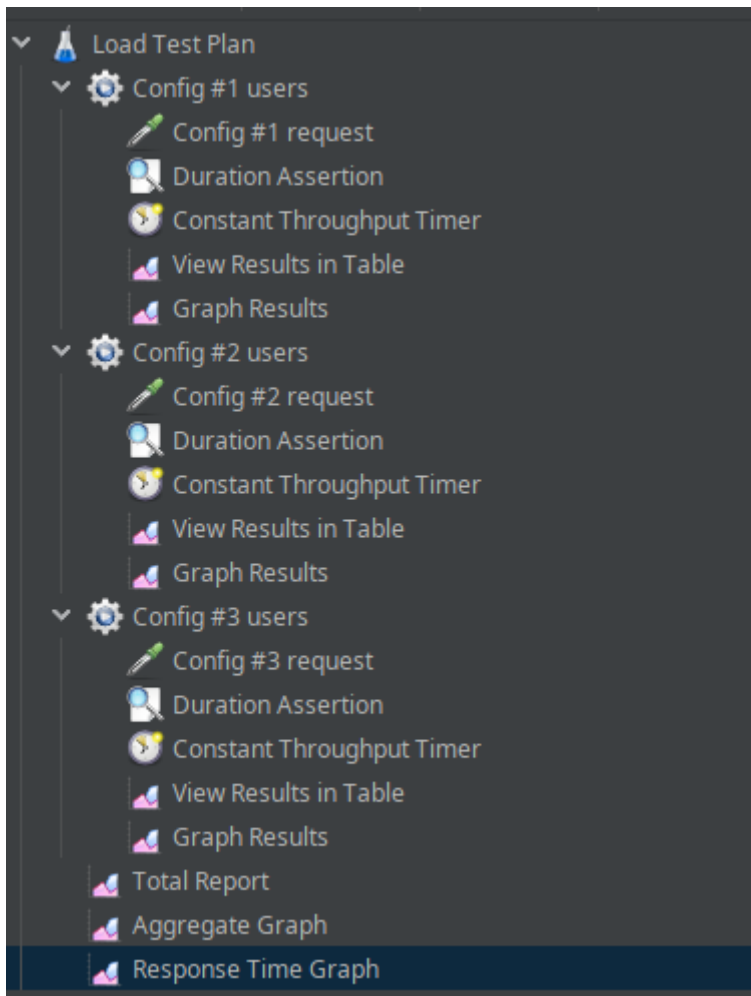


<https://github.com/maxbarsukov/itmo/tree/master/6%20%D1%82%D0%BF%D0%BE/%D0%BB%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D1%8B%D0%B5/lab4>

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).

Thread Group

Name:

Config #1 users

Comments:

Action to be taken after a Sampler error

☒ Continue

☐ Start Next Thread Loop

☐ Stop Thread

☐ Stop Test

☐ Stop Test Now

Thread Properties

Number of Threads (users):

12

Ramp-up period (seconds):

60

Loop Count:

☐ Infinite

20

☒ Same user on each iteration

☐ Delay Thread creation until needed

☐ Specify Thread lifetime

Duration (seconds):

Startup delay (seconds):

Также thread group содержит следующие под-объекты:

HTTP Request

осуществляет запросы к тестовому серверу

HTTP Request

Name:

Config #1 request

Comments:

Basic

Advanced

Web Server

Protocol (http):

http

Server Name or IP:

192.168.10.83

Port Number:

8080

HTTP Request

GET

Path:

/

Content encoding:

☐ Redirect Automatically

☒ Follow Redirects

☒ Use KeepAlive

☐ Use multipart/form-data

☐ Browser-compatible headers

Parameters

Body Data

Files Upload

Send Parameters With the Request:

Name:	Value	URL Encode?	Content-Type	Include Equals?
token	495351464	<input type="checkbox"/>	text/plain	<input checked="" type="checkbox"/>
user	-2105800574	<input type="checkbox"/>	text/plain	<input checked="" type="checkbox"/>
config	1	<input type="checkbox"/>	text/plain	<input checked="" type="checkbox"/>

Duration Assertion

осуществляет проверку времени отклика сервера

Duration Assertion

Name:

Comments:

- Apply to:

☐ Main sample and sub-samples ☒ Main sample only ☐ Sub-samples only

- Duration to Assert

Duration in milliseconds:

Constant Throughput Timer

определяет среднюю нагрузку, формируемую одним пользователем в мин

Constant Throughput Timer

Name:

Comments:

- Delay before each affected sampler

Target throughput (in samples per minute):

Calculate Throughput based on:

View Results in Table

Отображение результатов тестирования в виде таблицы

View Results in Table

Name:

Comments:

Write results to file / Read from file

Filename Log/Display Only: ☐ Errors ☐ Successes

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
----------	------------	-------------	-------	-----------------	--------	-------	------------	---------	------------------

Graph Results

Отображение результатов тестирования в виде графика

Graph Results

Name:

Graph Results

Comments:

Write results to file / Read from file

Filename

Browse...

Log/Display Only:

☐ Errors

☐ Successes

Configure

Graphs to Display

☒ Data

☒ Average

☒ Median

☒ Deviation

☒ Throughput

ms

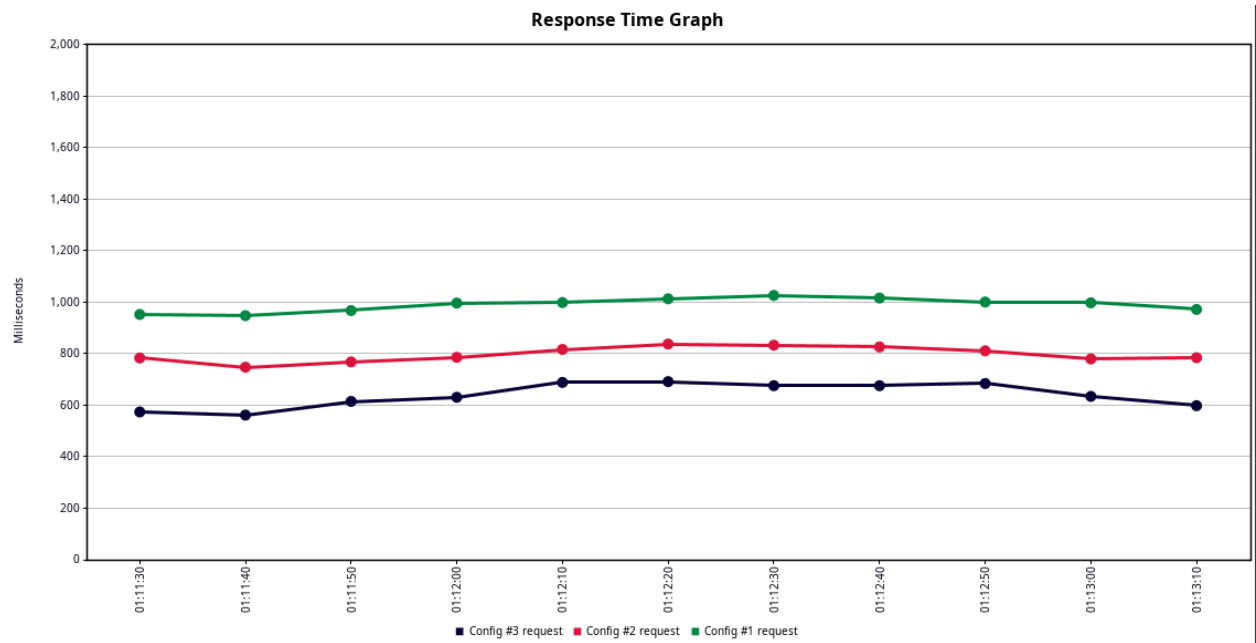
Запуск сценария

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

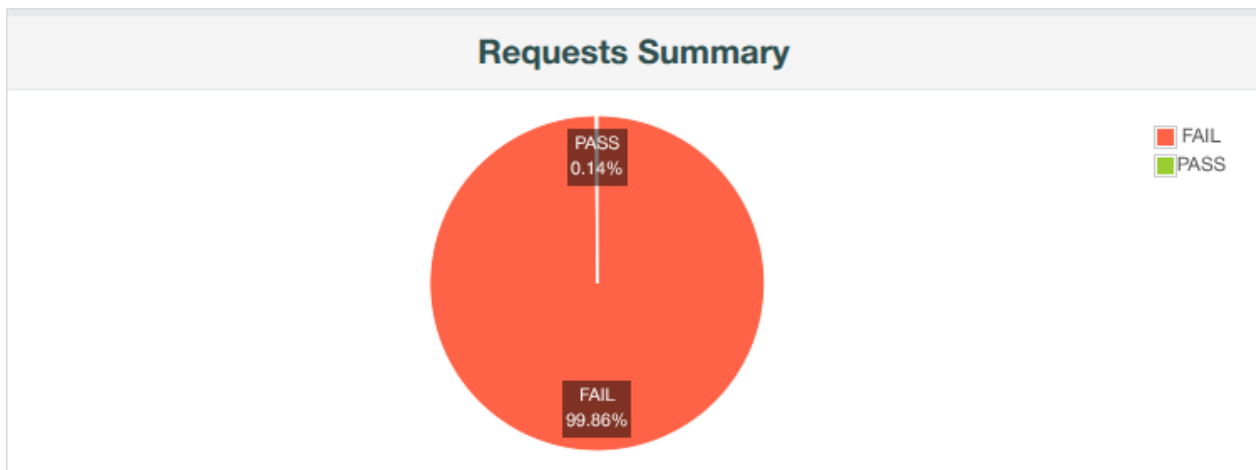
Результат:

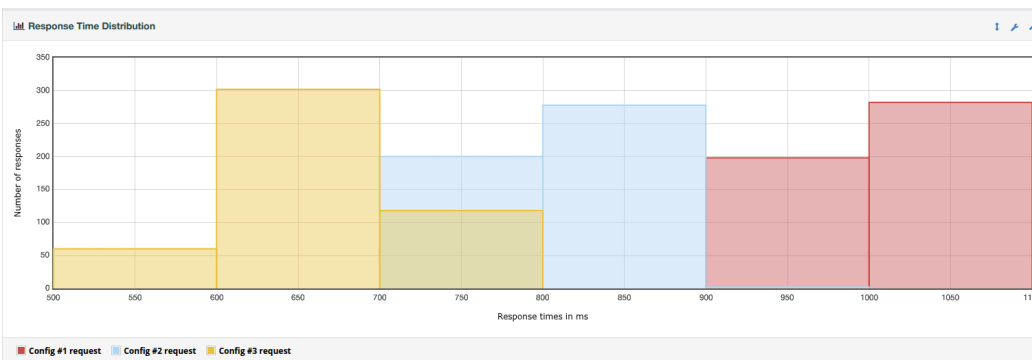
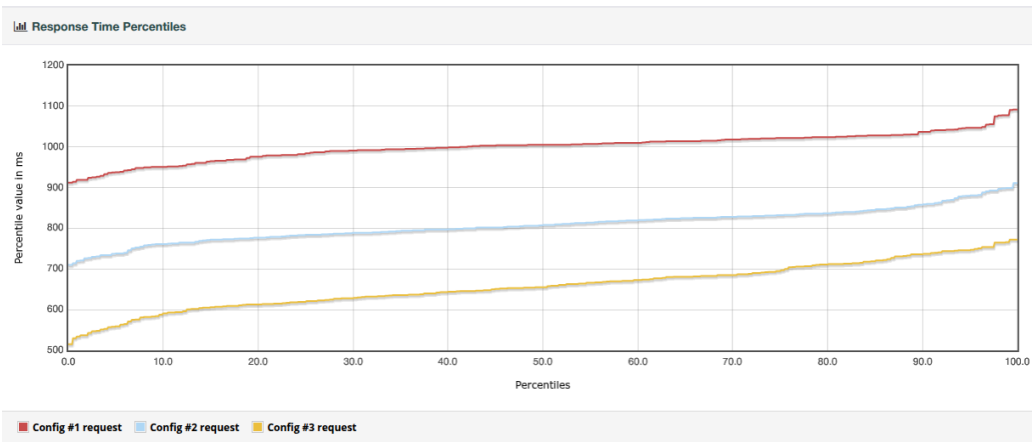
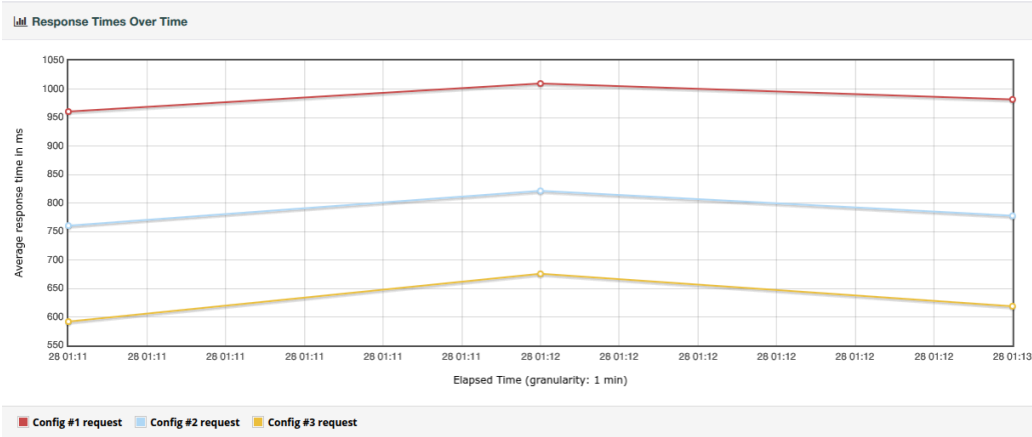
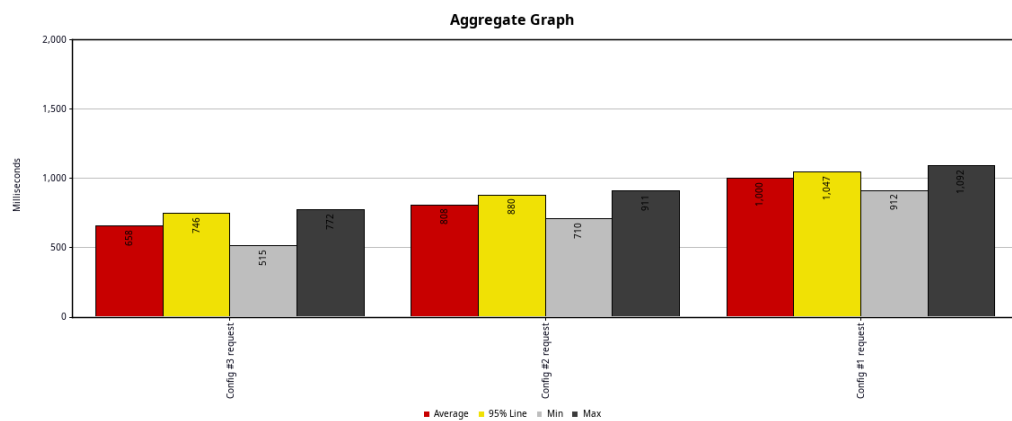
```
> jmeter -n -t ./load/test-plan.jmx -l ./load/result/results.csv -e -o ./load/report  
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release  
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release  
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release  
WARN StatusConsoleListener The use of package scanning to locate plugins is deprecated and will be removed in a future release  
Creating summariser <summary>  
Created the tree successfully using ./load/test-plan.jmx  
Starting standalone test @ 2025 May 28 01:11:35 GMT+03:00 (1748383895477)  
Waiting for possible Shutdown/StopTestNow/HeapDump/ThreadDump message on port 4445  
summary + 76 in 00:00:25 = 3.1/s Avg: 768 Min: 515 Max: 1055 Err: 75 (98.68%) Active: 15 Started: 15 Finished: 0  
summary + 255 in 00:00:30 = 8.5/s Avg: 830 Min: 581 Max: 1092 Err: 255 (100.00%) Active: 33 Started: 33 Finished: 0  
summary = 331 in 00:00:55 = 6.0/s Avg: 816 Min: 515 Max: 1092 Err: 330 (99.70%)  
summary + 285 in 00:00:30 = 9.5/s Avg: 840 Min: 608 Max: 1078 Err: 285 (100.00%) Active: 18 Started: 36 Finished: 18  
summary = 616 in 00:01:25 = 7.3/s Avg: 827 Min: 515 Max: 1092 Err: 615 (99.84%)  
summary + 104 in 00:00:28 = 3.7/s Avg: 793 Min: 534 Max: 1075 Err: 104 (100.00%) Active: 0 Started: 36 Finished: 36  
summary = 720 in 00:01:53 = 6.4/s Avg: 822 Min: 515 Max: 1092 Err: 719 (99.86%)  
Tidying up ... @ 2025 May 28 01:13:28 GMT+03:00 (1748384008473)  
... end of run
```

Графики пропускной способности приложения, полученные в ходе нагрузочного тестирования



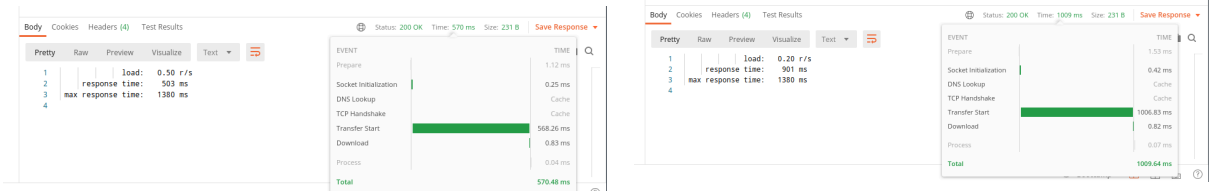
Statistics															
Requests	Executions			Response Times (ms)							Throughput	Network (KB/sec)			
Label	#Samples	FAIL	Error %	Average	Min	Max	Median	90th pct	95th pct	99th pct	Transactions/s	Received	Sent		
Total	1440	1438	99.86%	822.48	515	1092	807.50	1018.00	1028.00	1055.59	12.76	2.88	0.00		
Config #1 request	480	480	100.00%	1000.42	912	1092	1005.00	1037.00	1047.00	1080.47	4.25	0.96	0.00		
Config #2 request	480	480	100.00%	808.45	710	911	807.50	858.90	880.95	899.00	4.26	0.96	0.00		
Config #3 request	480	478	99.58%	658.57	515	772	655.00	736.90	746.95	767.14	4.27	0.96	0.00		





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:

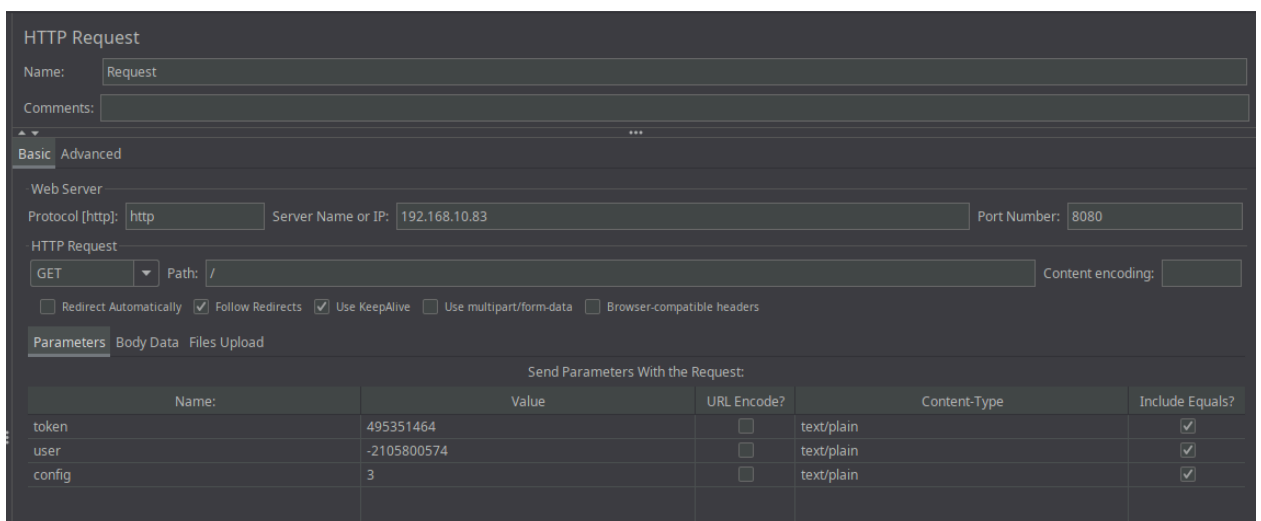
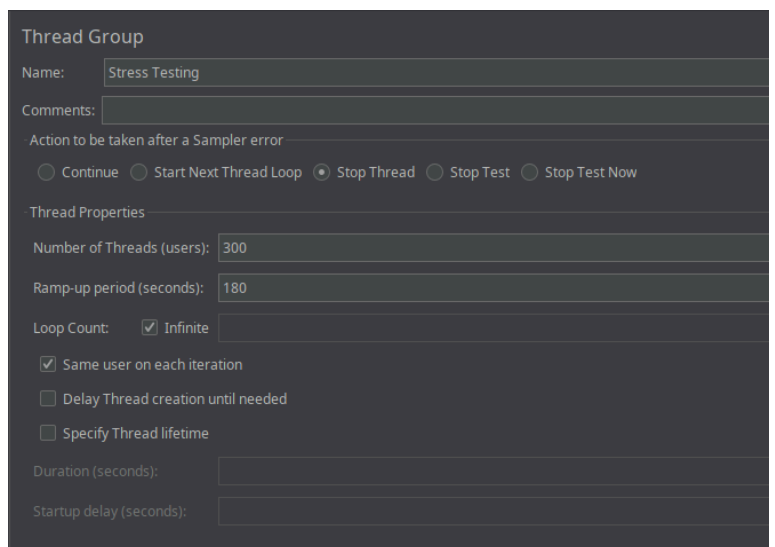
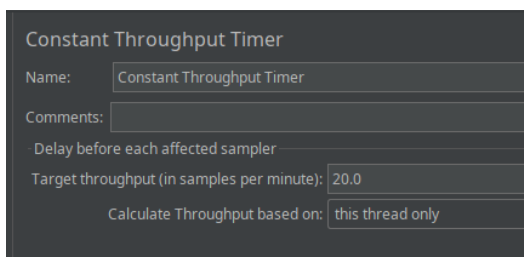
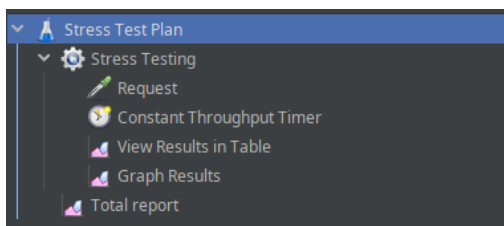
Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received KB...	Sent KB/sec
Config #3 r...	3	6458	6590	7192	7192	7192	5594	7192	100.00%	14.8/min	0.06	0.00
Config #2 r...	3	6460	6590	7196	7196	7196	5595	7196	100.00%	14.8/min	0.06	0.00
Config #1 r...	3	6465	6592	7208	7208	7208	5595	7208	100.00%	14.8/min	0.06	0.00
TOTAL	9	6461	6590	7196	7208	7208	5594	7208	100.00%	44.3/min	0.17	0.00

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



Send Parameters With the Request:				
Name:	Value	URL Encode?	Content-Type	Include Equals?
token	495351464	<input type="checkbox"/>	text/plain	<input checked="" type="checkbox"/>
user	-2105800574	<input type="checkbox"/>	text/plain	<input checked="" type="checkbox"/>
config	3	<input type="checkbox"/>	text/plain	<input checked="" type="checkbox"/>

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.

View Results in Table

Name: View Results in Table

Comments:

Write results to file / Read from file

Filename: /home/max/prog/itmo/itmo/6 тпо/лабораторные/lab4/stress/result/results.csv

Browse...

Log/Display Only: ☐ Errors ☐ Successes

Configure

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
7994	02:40:04.874	Stress Testing 1...	Request	2636	✓	231	162	2636	0
7995	02:40:04.874	Stress Testing 1...	Request	2636	✓	231	162	2636	0
7996	02:40:04.874	Stress Testing 1...	Request	2636	✓	231	162	2636	0
7997	02:40:04.874	Stress Testing 1...	Request	2636	✓	231	162	2636	0
7998	02:40:04.874	Stress Testing 1...	Request	2636	✓	231	162	2636	0
7999	02:40:04.876	Stress Testing 1...	Request	2634	✓	231	162	2634	0
8000	02:40:04.876	Stress Testing 1...	Request	2634	✓	231	162	2634	0
8001	02:40:04.876	Stress Testing 1...	Request	2634	✓	231	162	2634	0
8002	02:40:04.876	Stress Testing 1...	Request	2634	✓	231	162	2634	0
8003	02:40:05.513	Stress Testing 1...	Request	2002	✓	231	162	2002	0
8004	02:40:04.876	Stress Testing 1...	Request	2639	✓	231	162	2639	0
1879	02:36:16.909	Stress Testing 1...	Request	5562	✗	233	162	5562	11
1880	02:36:17.735	Stress Testing 1...	Request	4736	✗	233	162	4736	7
1881	02:36:18.633	Stress Testing 1...	Request	3838	✗	233	162	3838	0
1882	02:36:18.336	Stress Testing 1...	Request	4135	✗	233	162	4135	4
1883	02:36:17.135	Stress Testing 1...	Request	5336	✗	233	162	5336	4
1970	02:36:21.225	Stress Testing 1...	Request	4956	✗	233	162	4956	0
1971	02:36:21.225	Stress Testing 1...	Request	4956	✗	233	162	4956	0
1972	02:36:21.225	Stress Testing 1...	Request	4956	✗	233	162	4956	0
1973	02:36:21.225	Stress Testing 1-7	Request	4956	✗	233	162	4956	0
1975	02:36:21.225	Stress Testing 1...	Request	4957	✗	233	162	4957	0
1976	02:36:21.226	Stress Testing 1...	Request	4956	✗	233	162	4956	0
1977	02:36:21.226	Stress Testing 1-1	Request	4956	✗	233	162	4956	0
1978	02:36:21.226	Stress Testing 1...	Request	4956	✗	233	162	4956	0

☐ Scroll automatically?

☐ Child samples?

No of Samples 8085

Latest Sample 1696

Average 3081

Deviation 1591

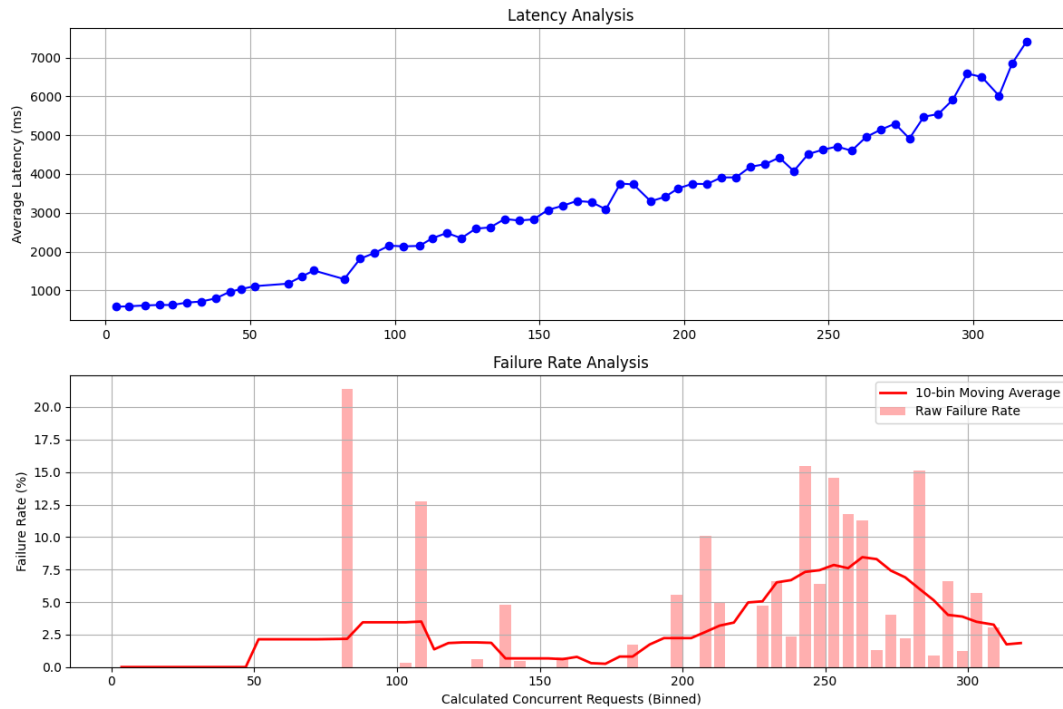
# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput
8085	3481	3271	5674	6515	7923	14	9229	3.71%	25.4/sec
8085	3481	3271	5674	6515	7923	14	9229	3.71%	25.4/sec

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

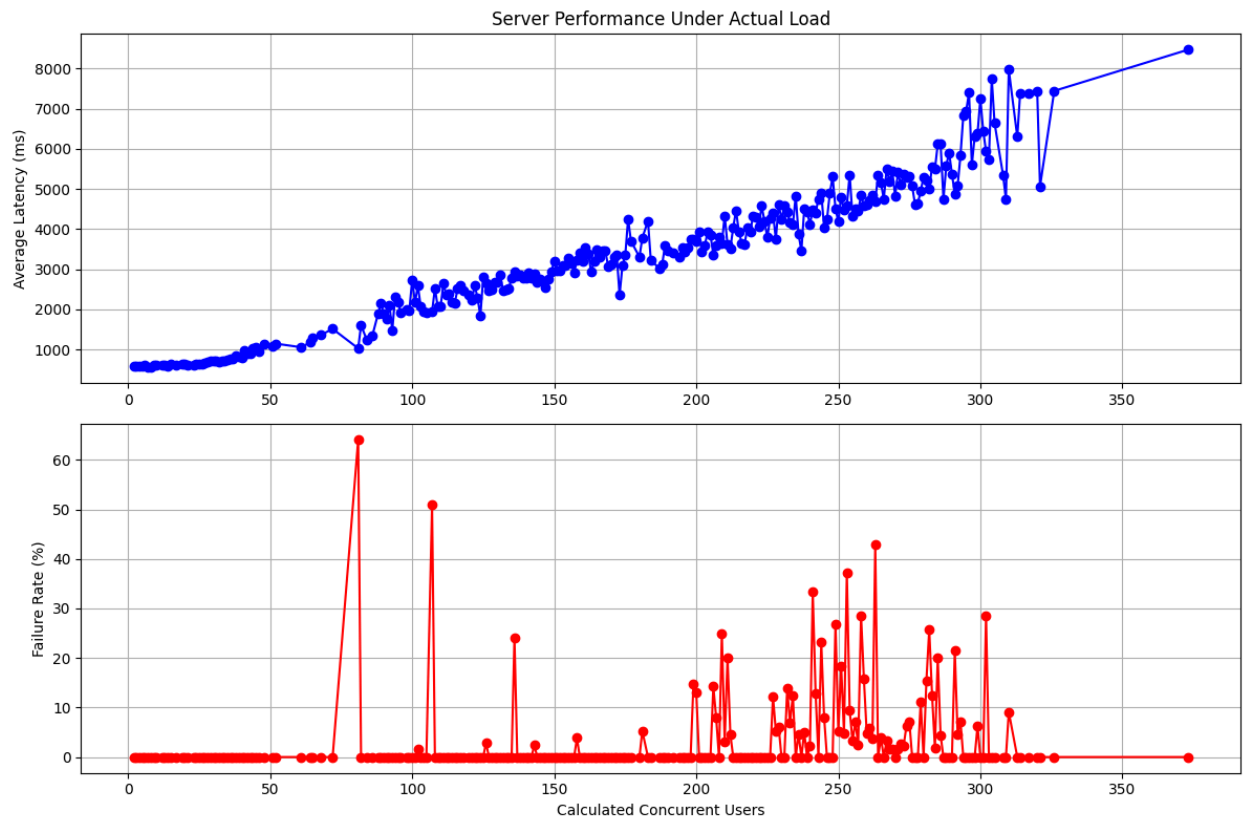
Попробуем проанализировать полученные результаты:

First failures observed at 81 concurrent users

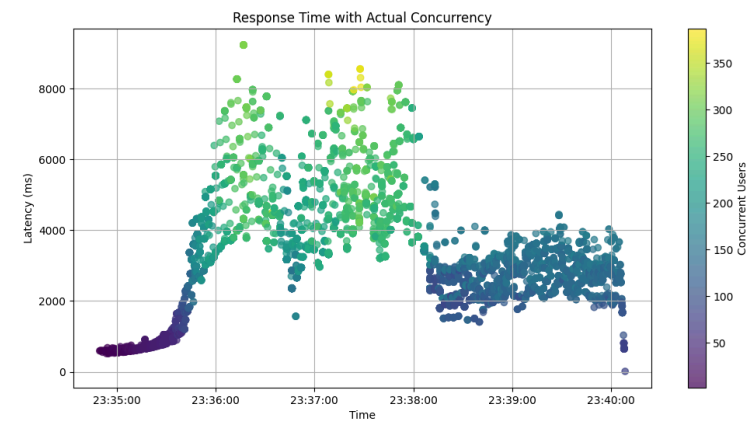
Оценка задержки и частоты ошибок



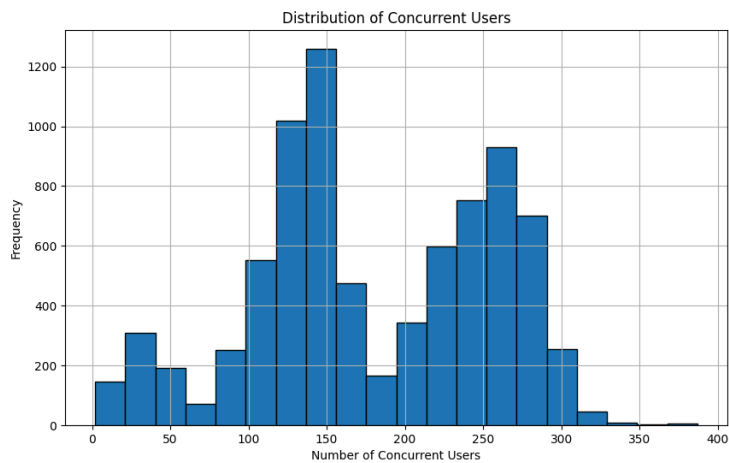
Оценка средней задержки для кол-ва параллельных пользователей



Оценка времени отклика во времени от кол-ва параллельных пользователей



Распределение параллельных пользователей



Перцентили времени обработки запроса во времени

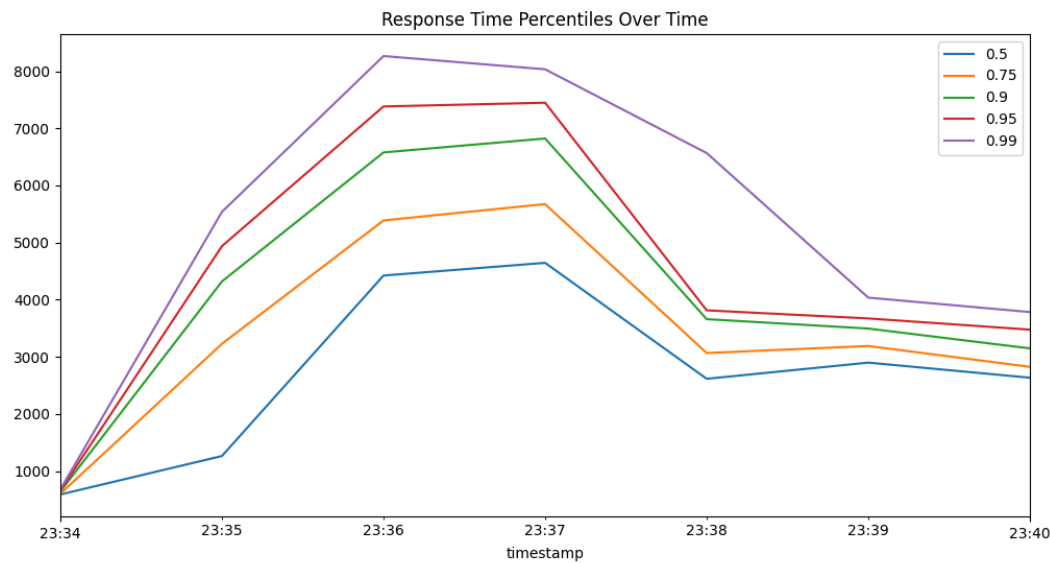
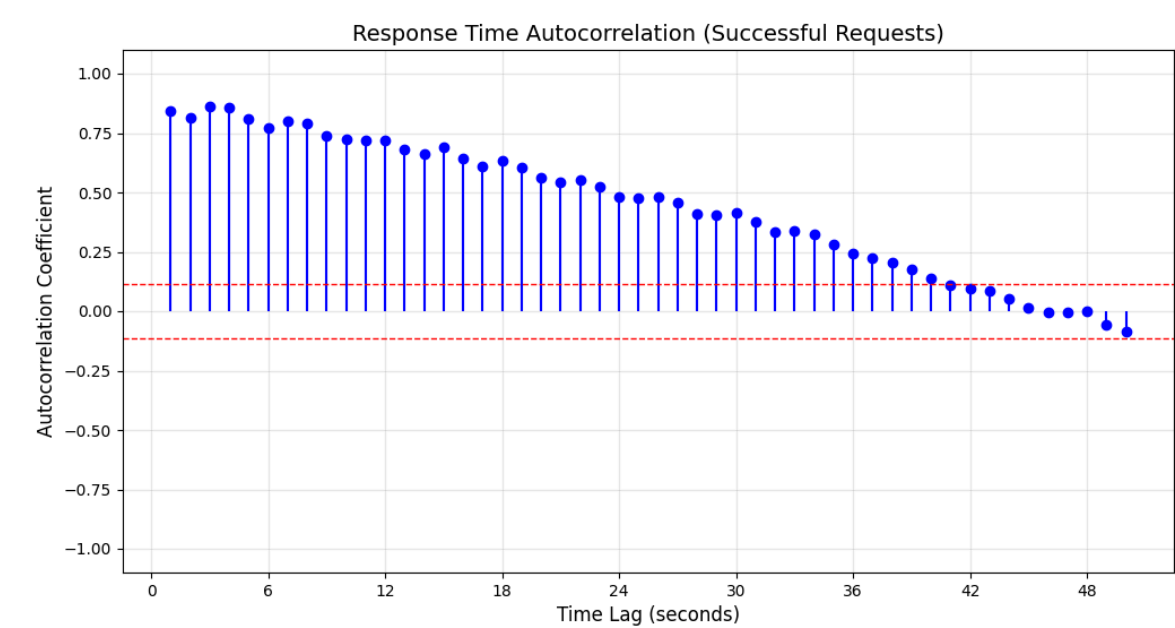


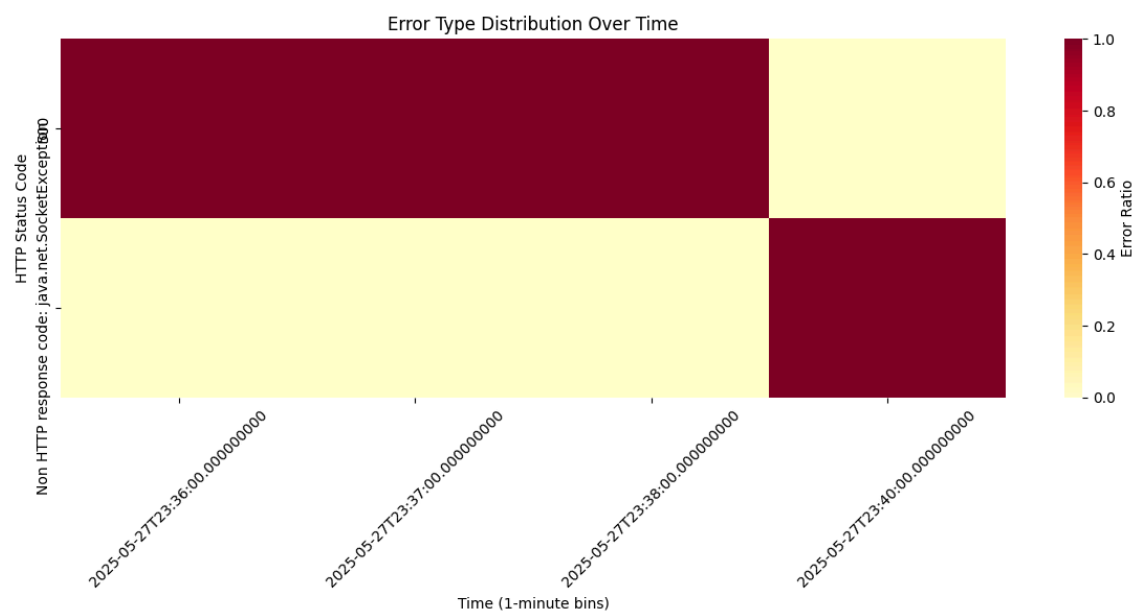
График автокорреляции для времени отклика



- 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

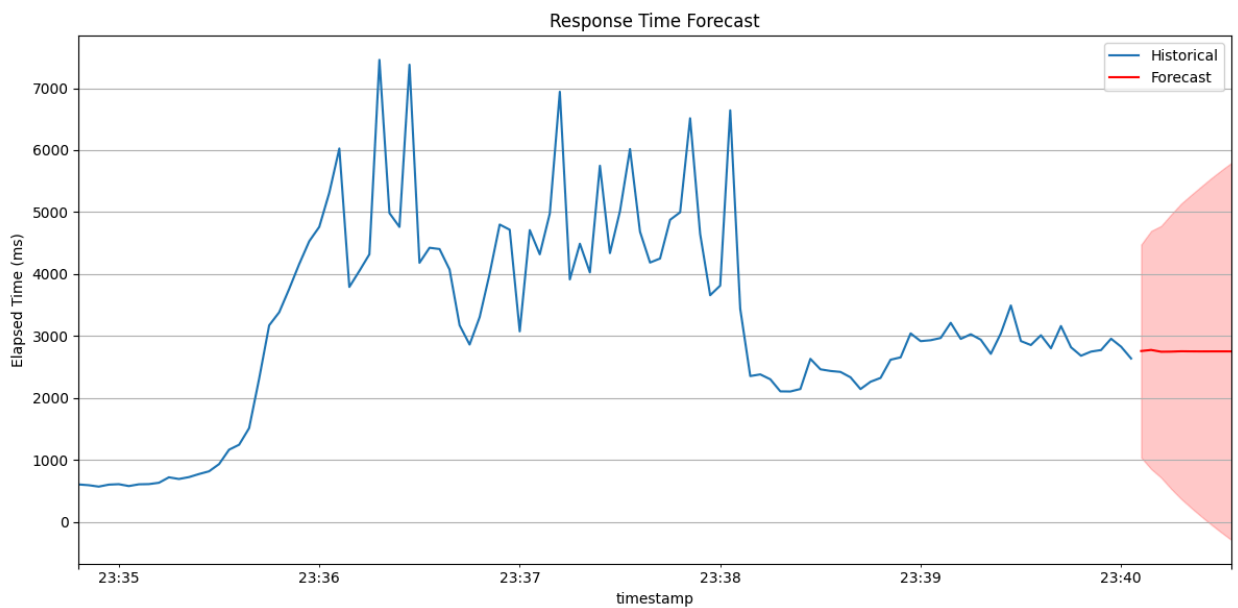


Кластеризация ошибок во временном пространстве



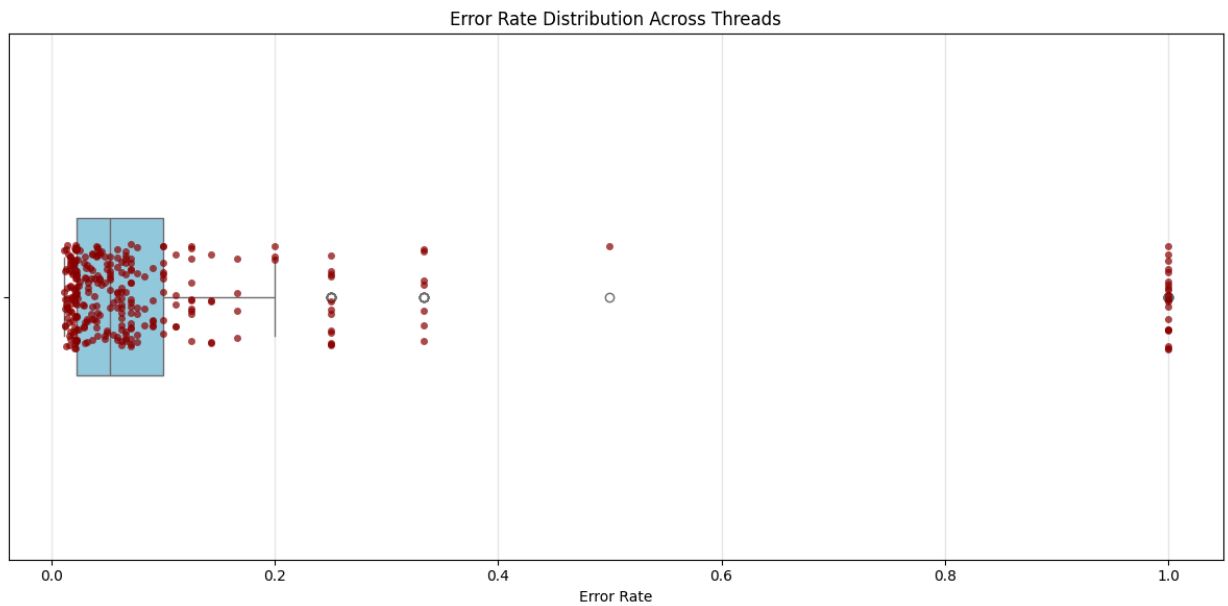
- Крупные кластеры (много точек одного цвета) → периоды нестабильности системы.
- Интервалы между кластерами → периоды стабильной работы
- Кластеры ошибок совпадают ли с пиками нагрузки.

Прогнозирование деградации времени отклика



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

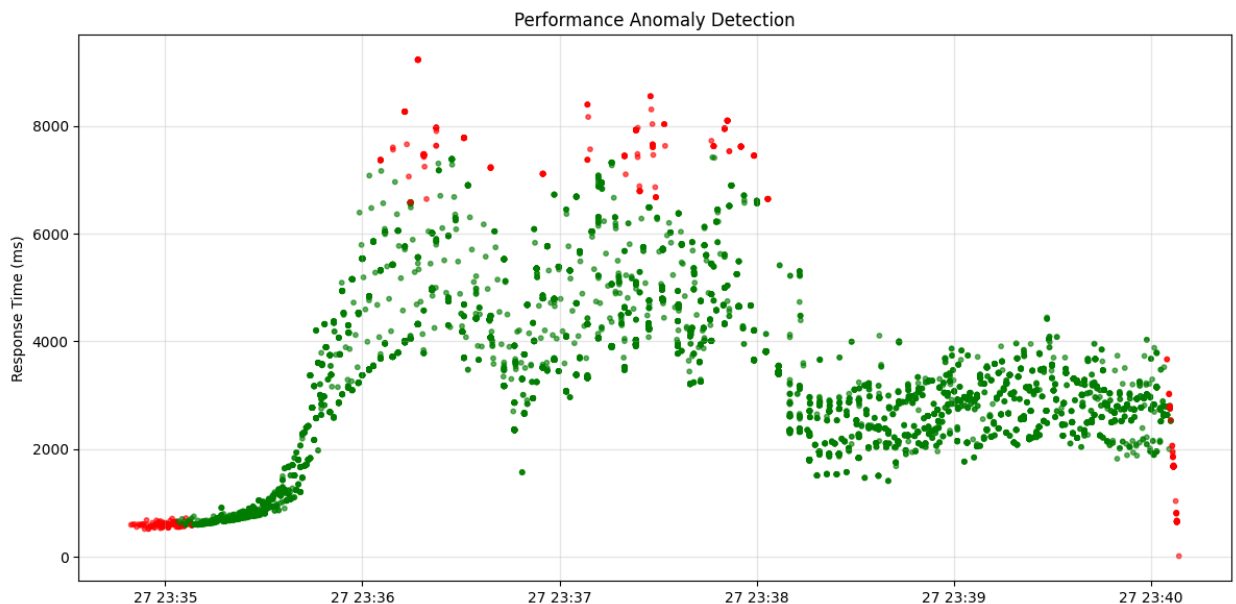
Распределение частоты ошибок между потоками



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

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

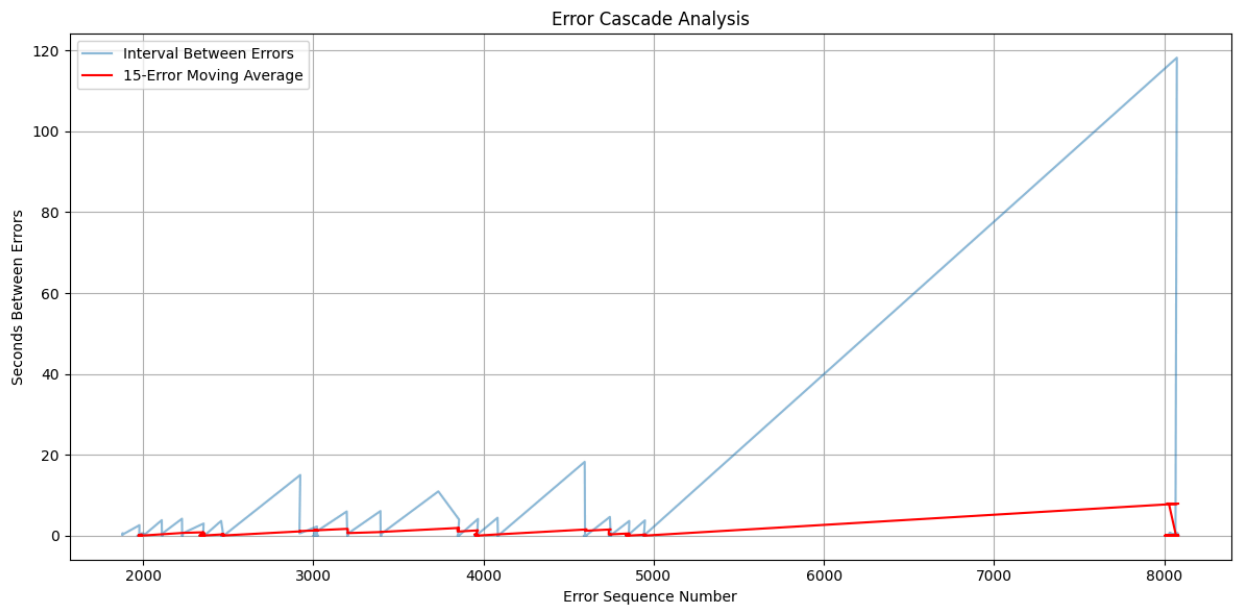
Выявление аномалий в производительности



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 требованиям по максимальному времени отклика. Для этого был построен график зависимости времени отклика приложения от нагрузки.