

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И
КОМПЬЮТЕРНОЙ ТЕХНИКИ

ЛАБОРАТОРНАЯ РАБОТА №4
по дисциплине
«Тестирование программного обеспечения»

Вариант №3409

Выполнил:
Студент группы Р3334
Баянов Равиль
Динарович
Преподаватель:
Бострикова Дарья
Константиновна

Санкт-Петербург
2025

Оглавление

Задание.....	3
Описание конфигурации JMeter для нагрузочного тестирования	4
Графики пропускной способности приложения, полученные в ходе нагрузочного тестирования.....	5
Выводы по выбранной конфигурации аппаратного обеспечения	6
График изменения времени отклика от нагрузки для выбранной конфигурации, полученный в ходе стресс-тестирования системы.....	8
Вывод.....	9

Задание

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.

Приложение для тестирования доступно только во внутренней сети кафедры.

Если запрос содержит некорректные параметры, сервер возвращает HTTP 403.

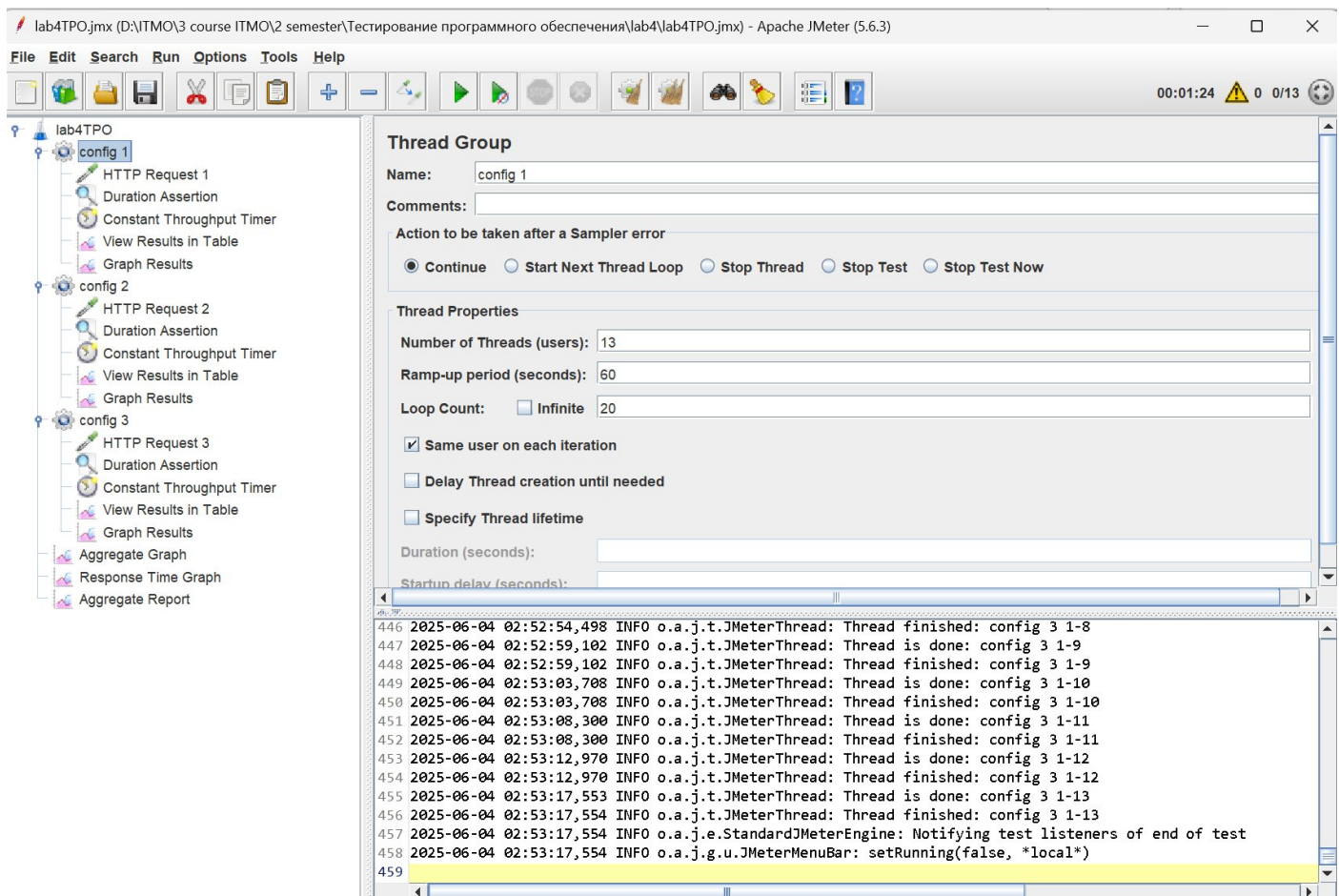
Если приложение не справляется с нагрузкой, сервер возвращает HTTP 503.

Параметры тестируемого веб-приложения:

- URL первой конфигурации (\$ 2300)
 - <http://stload.se.ifmo.ru:8080?token=495351469&user=-2105057934&config=1;>
- URL второй конфигурации (\$ 3100)
 - <http://stload.se.ifmo.ru:8080?token=495351469&user=-2105057934&config=2;>
- URL третьей конфигурации (\$ 4200)
 - <http://stload.se.ifmo.ru:8080?token=495351469&user=-2105057934&config=3;>
- Максимальное количество параллельных пользователей - 13;
- Средняя нагрузка, формируемая одним пользователем - 40 запр. в мин.;
- Максимально допустимое время обработки запроса - 640 мс.

Описание конфигурации JMeter для нагрузочного тестирования

Весь проект делится на 3 конфигурации, они отличаются между собой только параметром `config`, который мы будем передавать в GET запросе. В разделах *config* указывается общая информация о тесте, такая как: общее количество пользователей (поток), Ramp-up period, который указывает нам в течение, какого времени запустятся все потоки и Loop Count – кол-во тестов, которое мы хотим запустить.



Затем, в каждой конфигурации отдельно настраиваются samplers, timers, assertions и listeners.

Http Request – запросы к самому серверу (sampler).

Здесь указывается порт, который мы слушаем и указываем сервер, которому будем посылать запросы.

HTTP Request

Name:

Comments:

Basic **Advanced**

Web Server
 Protocol [http]: Server Name or IP: Port Number:

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	495351469	<input type="checkbox"/>	text/plain	<input checked="" type="checkbox"/>
user	2105057934	<input type="checkbox"/>	text/plain	<input checked="" type="checkbox"/>
config	1	<input type="checkbox"/>	text/plain	<input checked="" type="checkbox"/>

Duration Assertions – порог, который помогает отслеживать допустимое время отклика сервера (Assertion).

Duration Assertion Clear

Name:

Comments:

Apply to:
☐ Main sample and sub-samples
 ☒ Main sample only
 ☐ Sub-samples only

Duration to Assert
 Duration in milliseconds:

Constant Throughput Timer – это таймер, который позволяет приблизить значение пропускной способности всего теста к определенному значению (timer).

Constant Throughput Timer

Name:

Comments:

Delay before each affected sampler

Target throughput (in samples per minute):

Calculate Throughput based on: ▼

View Results in Table – это компонент, позволяющий видеть обработку запросов в виде таблицы (listener).

View Results in Table

Name:

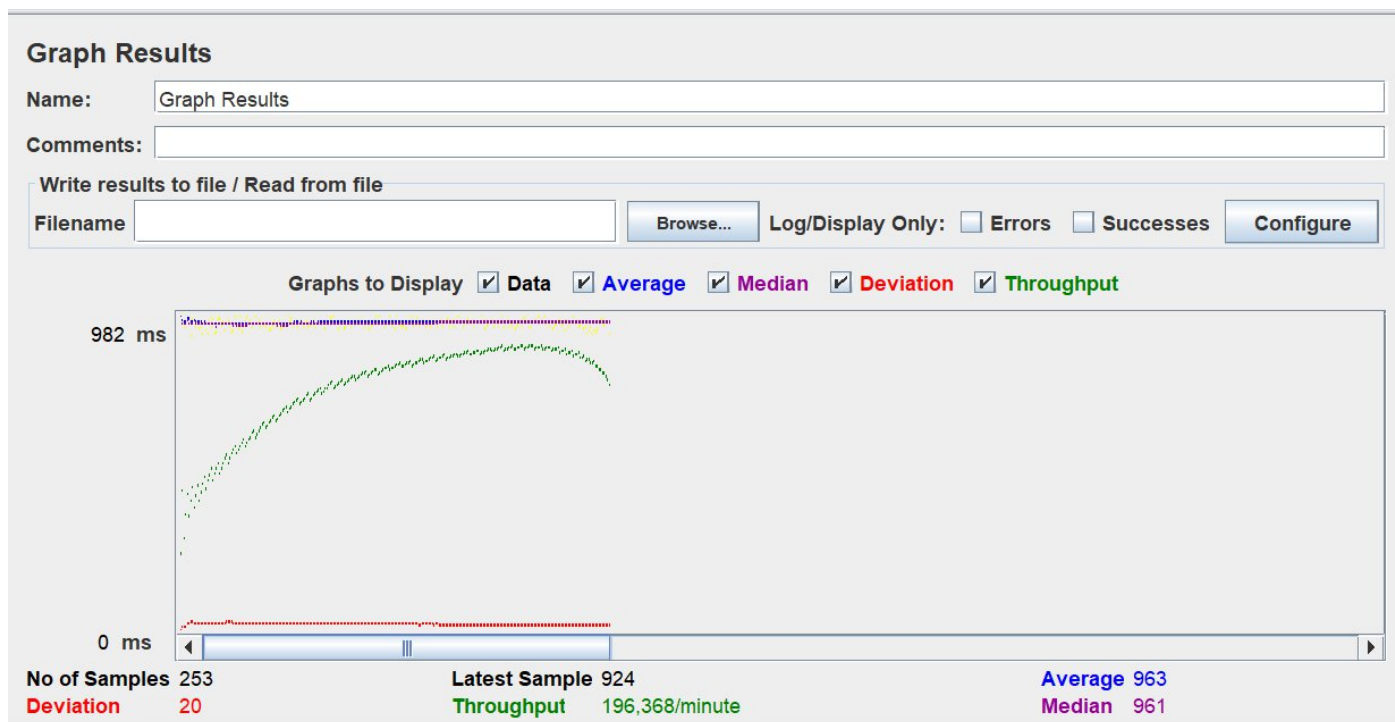
Comments:

Write results to file / Read from file

Filename Log/Display Only: ☐ Errors ☐ Successes

Sample #	Start Time	Thread Name	Label	Sample Tim...	Status	Bytes	Sent Bytes	Latency	Connect Ti..
231	02:49:19.994	config 1 1-11	HTTP Requ...	947	✖	231	158	947	
232	02:49:20.109	config 1 1-12	HTTP Requ...	956	✖	231	158	956	
233	02:49:20.225	config 1 1-13	HTTP Requ...	965	✖	231	158	965	
234	02:49:21.380	config 1 1-10	HTTP Requ...	979	✖	231	158	979	
235	02:49:21.494	config 1 1-11	HTTP Requ...	957	✖	231	158	957	
236	02:49:21.610	config 1 1-12	HTTP Requ...	966	✖	231	158	966	
237	02:49:21.725	config 1 1-13	HTTP Requ...	946	✖	231	158	946	
238	02:49:22.994	config 1 1-11	HTTP Requ...	955	✖	231	158	955	
239	02:49:23.110	config 1 1-12	HTTP Requ...	961	✖	231	158	961	
240	02:49:23.225	config 1 1-13	HTTP Requ...	940	✖	231	158	940	
241	02:49:24.494	config 1 1-11	HTTP Requ...	950	✖	231	158	950	
242	02:49:24.609	config 1 1-12	HTTP Requ...	928	✖	231	158	928	
243	02:49:24.727	config 1 1-13	HTTP Requ...	937	✖	231	158	937	
244	02:49:25.994	config 1 1-11	HTTP Requ...	948	✖	231	158	948	

Graph Result – компонент, который позволяет увидеть результаты тестирования в виде графика.



Также в конфигурации присутствуют компонент для оценки общей картины тестирования, они помогают сравнить 3 конфигурации между собой.

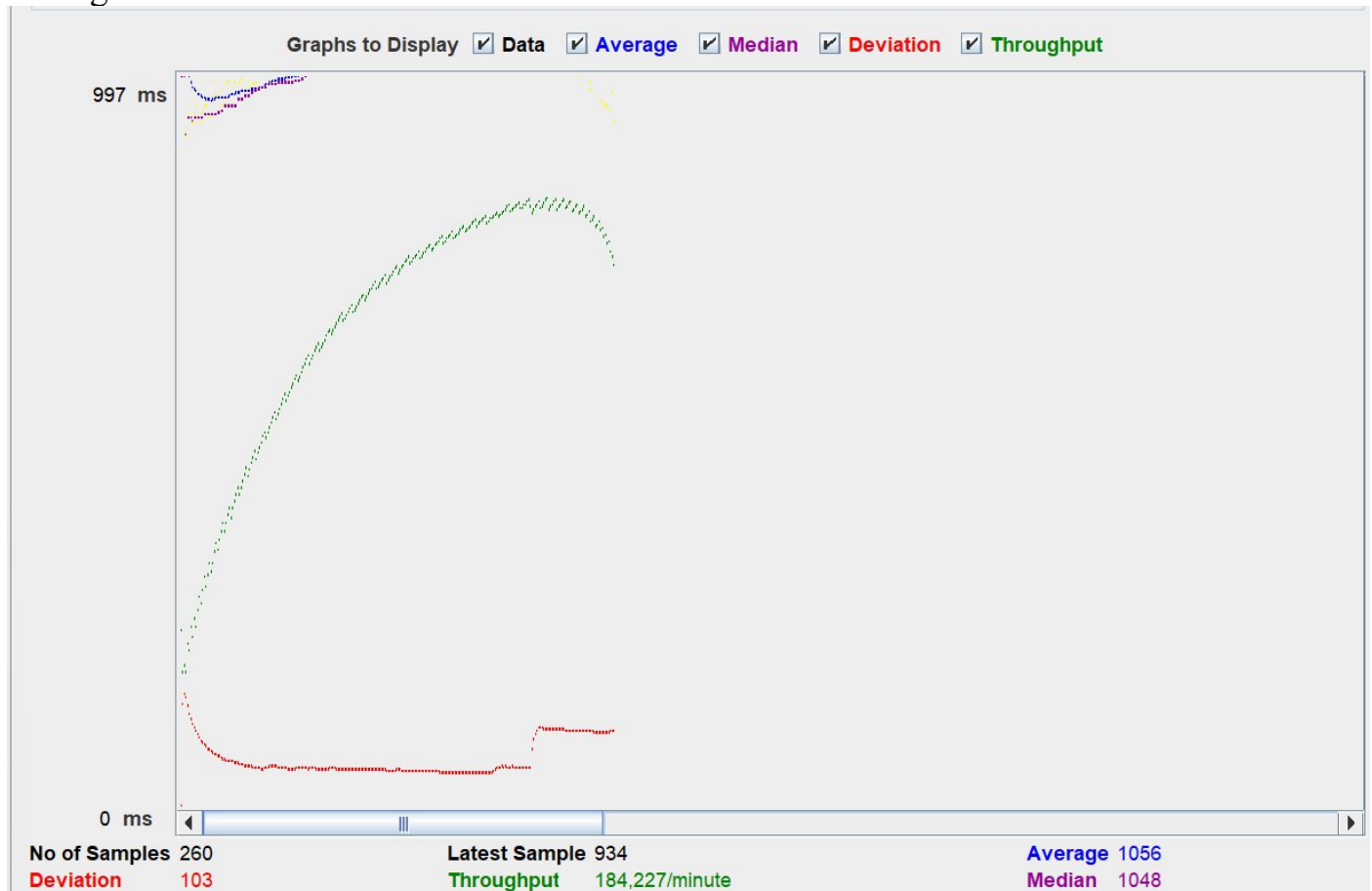
Response Time Graph – компонент, который строит график зависимости времени отклика от всего времени.

Aggregate Graph – компонент, который позволяет увидеть график минимального, максимального, среднего значения наших конфигураций по времени отклика.

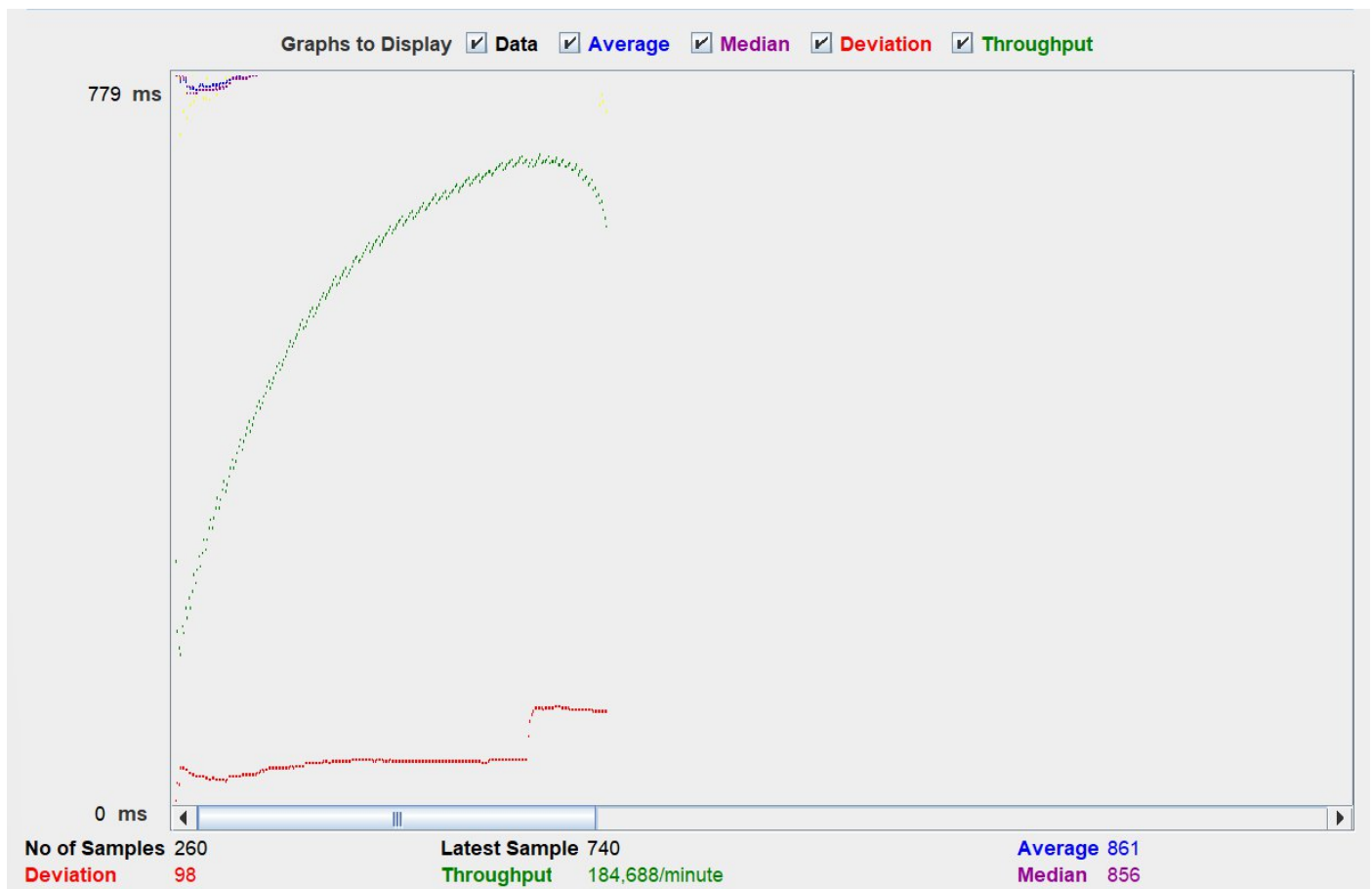
Aggregate Report – то, же что и aggregate graph, только позволяет нам увидеть результат в табличном виде и сохранить его в csv формат.

Графики пропускной способности приложения, полученные в ходе нагрузочного тестирования

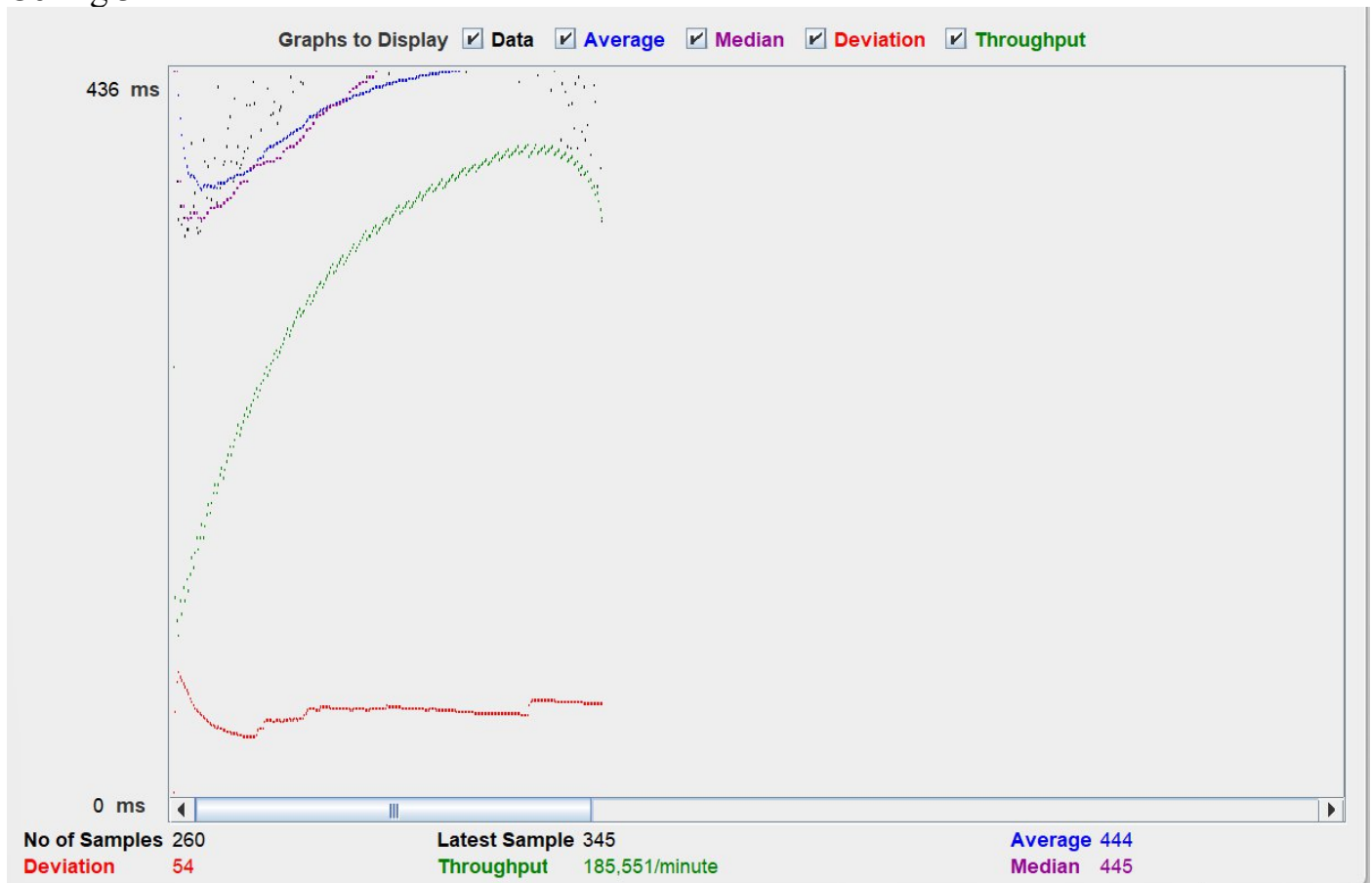
Config 1



Config 2



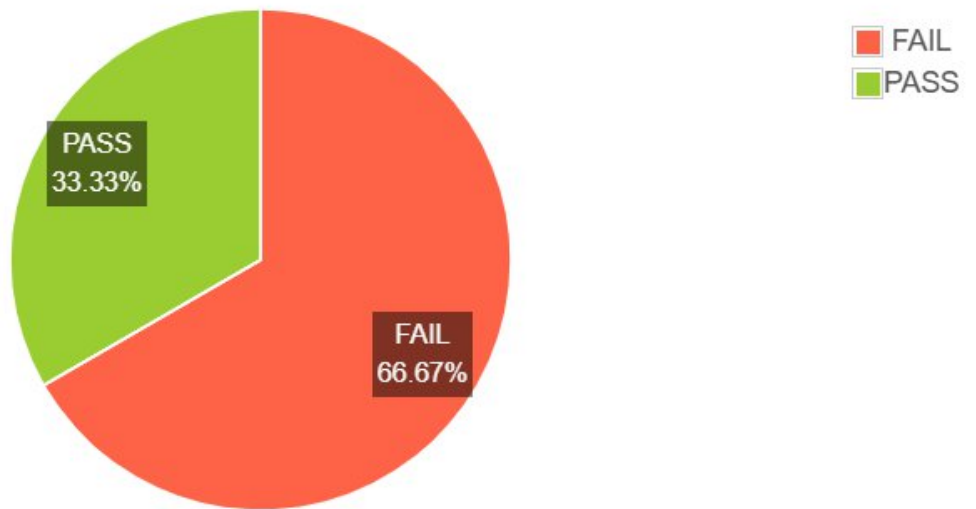
Config 3



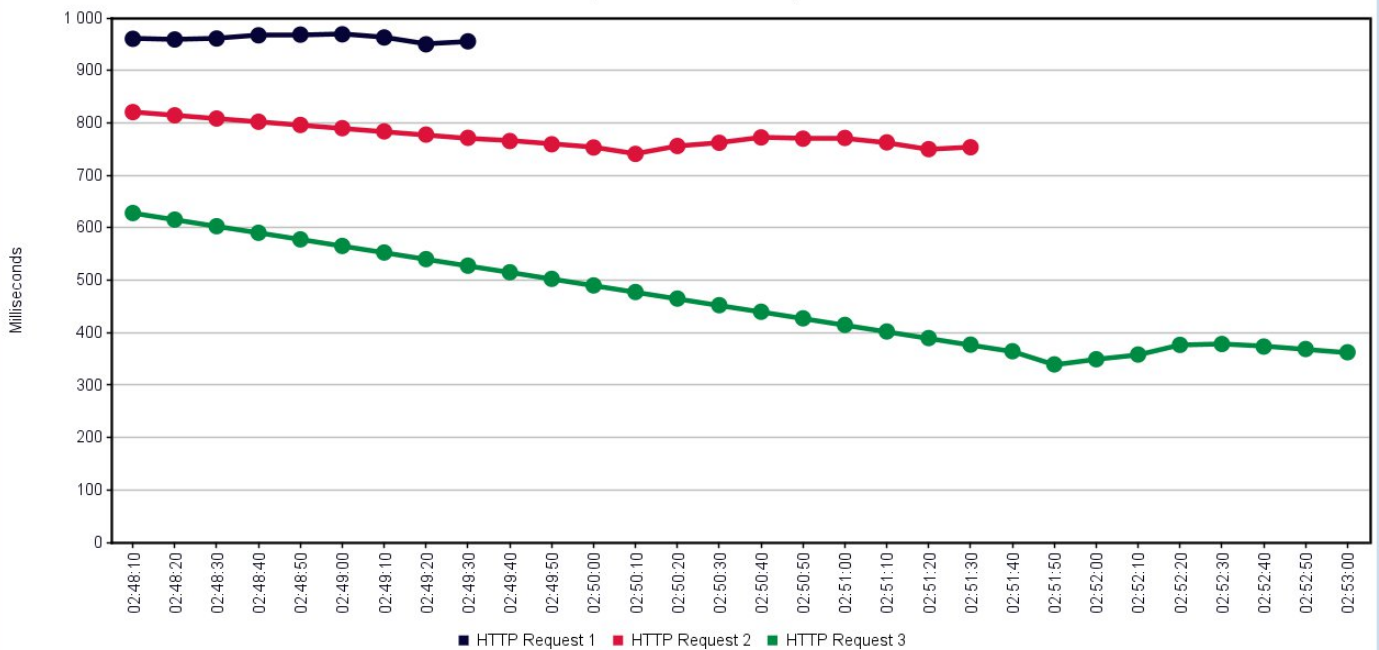
Statistics

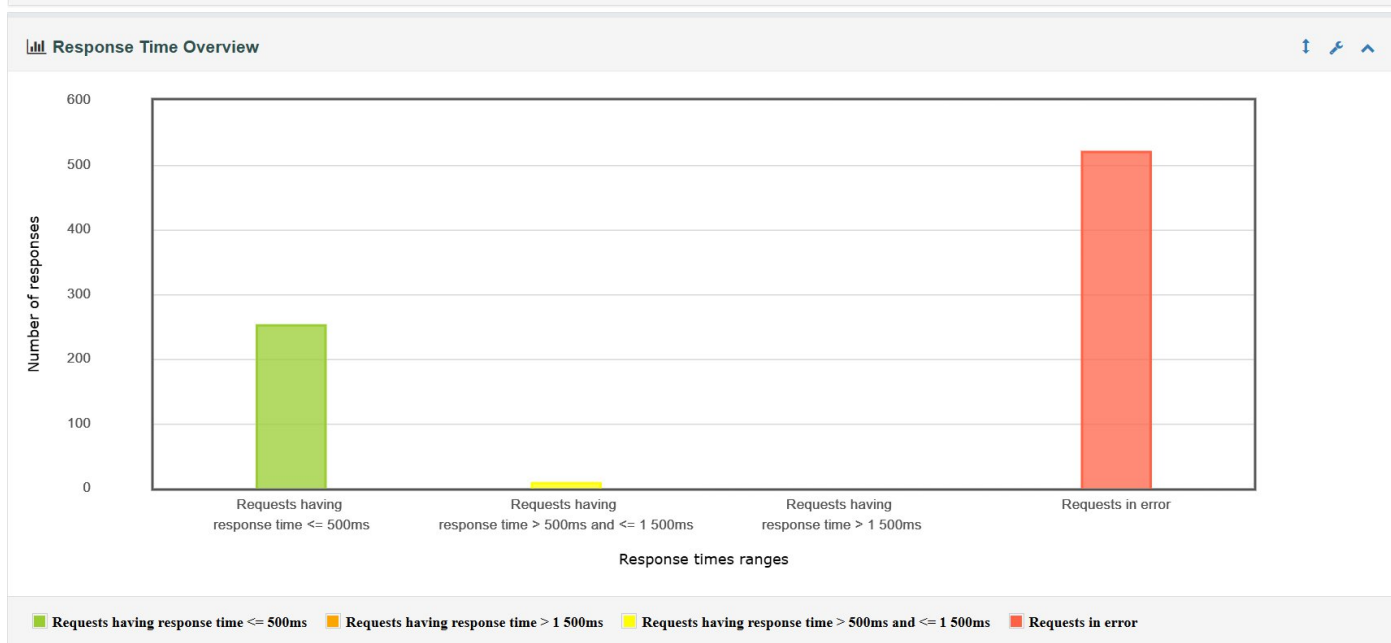
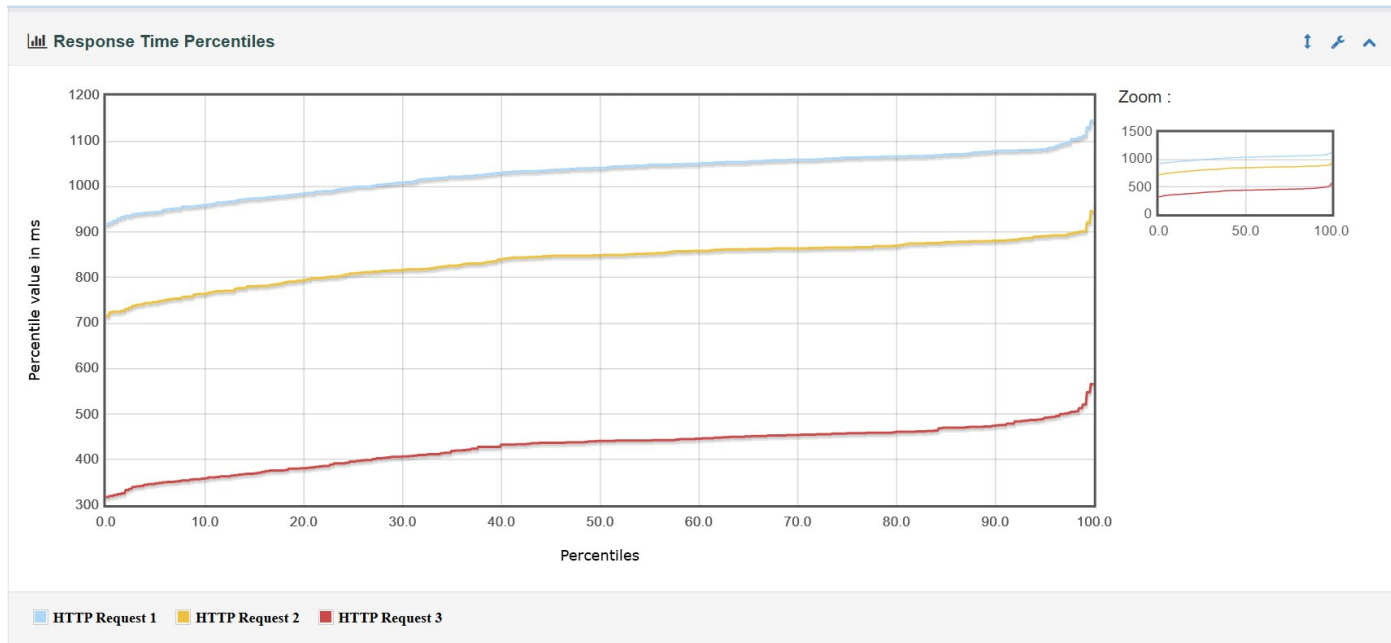
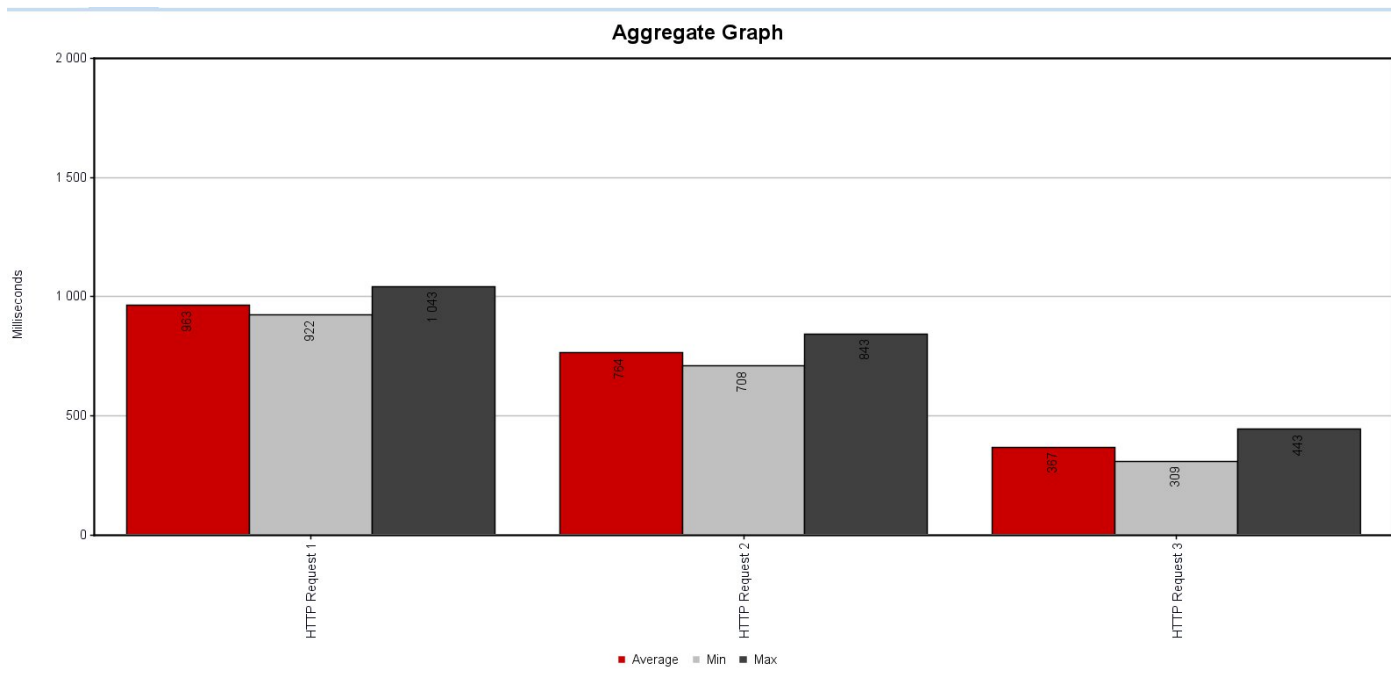
Requests	Executions			Response Times (ms)							Throughput	Network (KB/sec)	
Label	#Samples	FAIL	Error %	Average	Min	Max	Median	90th pct	95th pct	99th pct	Transactions/s	Received	Sent
Total	780	520	66.67%	764.33	318	1145	849.00	1059.00	1070.00	1095.38	9.21	2.08	1.42
HTTP Request 1	260	260	100.00%	1029.63	918	1145	1041.00	1078.00	1082.00	1119.02	3.07	0.69	0.47
HTTP Request 2	260	260	100.00%	835.63	716	946	849.00	881.00	891.00	909.41	3.08	0.69	0.47
HTTP Request 3	260	0	0.00%	427.73	318	566	441.00	474.90	491.85	531.53	3.09	0.70	0.48

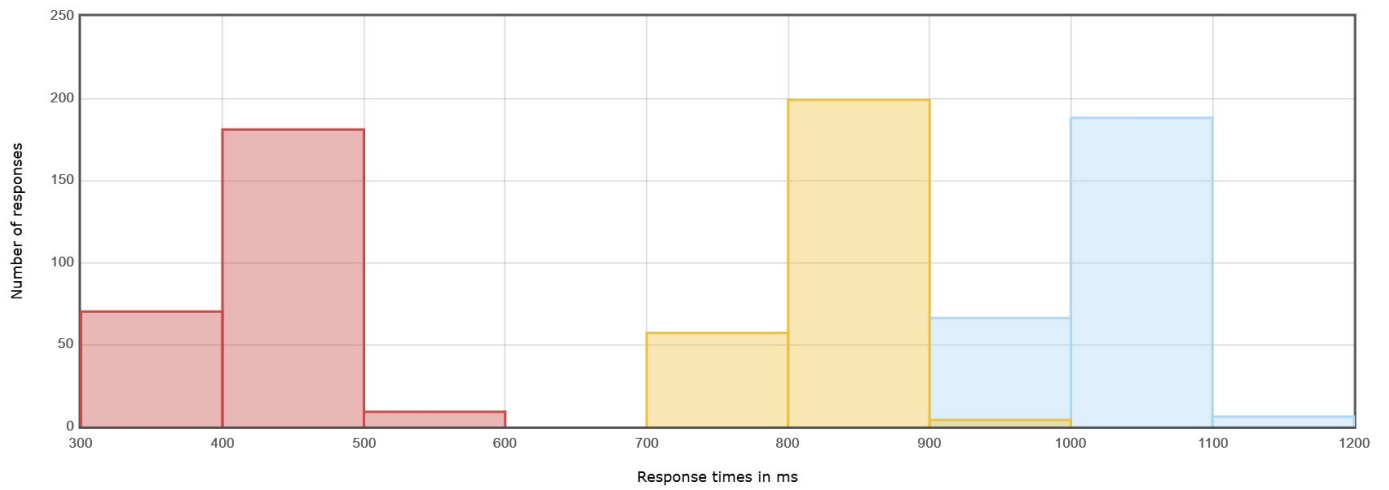
Requests Summary



Response Time Graph







HTTP Request 1 HTTP Request 2 HTTP Request 3

Выводы по выбранной конфигурации аппаратного обеспечения

Проанализировав данные графики, можем с уверенностью сказать, что нам подходит только конфигурация под номером 3, так как именно в ней мы не получили ни одной ошибки при тестировании. При этом конфигурация 2 в целом почти уложилось в нужный нам диапазон времени отклика, но всё равно недостаточно. Заметим, также что в 3 конфигурации сильно падает пропускная способность по сравнению с другими конфигурациями, но это не мешает нам выбрать 3 конфигурацию так как, пропускная способность 2 и 1 конфигурации хоть и постоянная, но она всё же не укладывается в рамки нашего варианта.

Описание конфигурации JMeter для стресс-тестирования

Для стресс-тестирования остановимся на 3 конфигурации, так как только она удовлетворяет всем нужным условиям. Попробуем увеличить количество пользователей до 300, уберём assertion, а всё остальное останется таким же.

The screenshot shows the 'HTTP Request' configuration dialog in JMeter. The 'Name' field is set to 'HTTP Request'. The 'Basic' tab is selected, showing the 'Web Server' section with 'Protocol' set to 'http', 'Server Name or IP' set to 'localhost', and 'Port Number' set to '34543'. The 'HTTP Request' section shows the method set to 'GET' and the path set to '/'. Below this, there are checkboxes for 'Redirect Automatically' (unchecked), 'Follow Redirects' (checked), 'Use KeepAlive' (checked), 'Use multipart/form-data' (unchecked), and 'Browser-compatible headers' (unchecked). The 'Parameters' tab is also visible, showing a table for 'Send Parameters With the Request:' with columns for 'Name:', 'Value', 'URL Encode?', 'Content-Type', and 'Include Equals?'. At the bottom, there are buttons for 'Detail', 'Add', 'Add from Clipboard', 'Delete', 'Up', and 'Down'.

HTTP Request

Name: HTTP Request

Comments:

Basic **Advanced**

Web Server

Protocol [http]: http Server Name or IP: localhost Port Number: 34543

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?
-------	-------	-------------	--------------	-----------------

Detail Add Add from Clipboard Delete Up Down

Constant Throughput Timer

Name: Constant Throughput Timer

Comments:

Delay before each affected sampler

Target throughput (in samples per minute): 40.0

Calculate Throughput based on: this thread only

Thread Group

Name: config 3

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): 300

Ramp-up period (seconds): 2

Loop Count: ☐ Infinite 1

☒ Same user on each iteration

☐ Delay Thread creation until needed

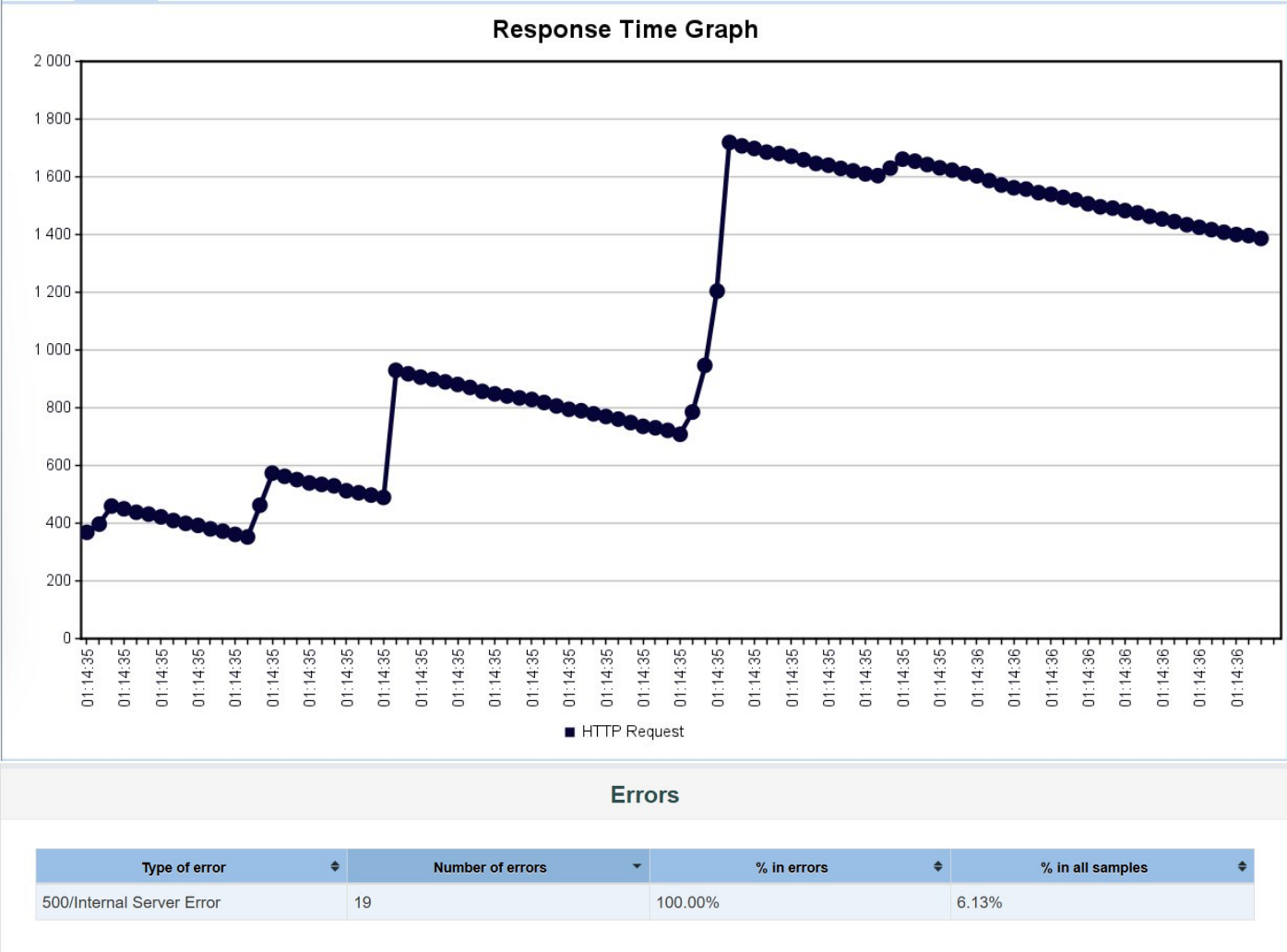
☐ Specify Thread lifetime

Duration (seconds):

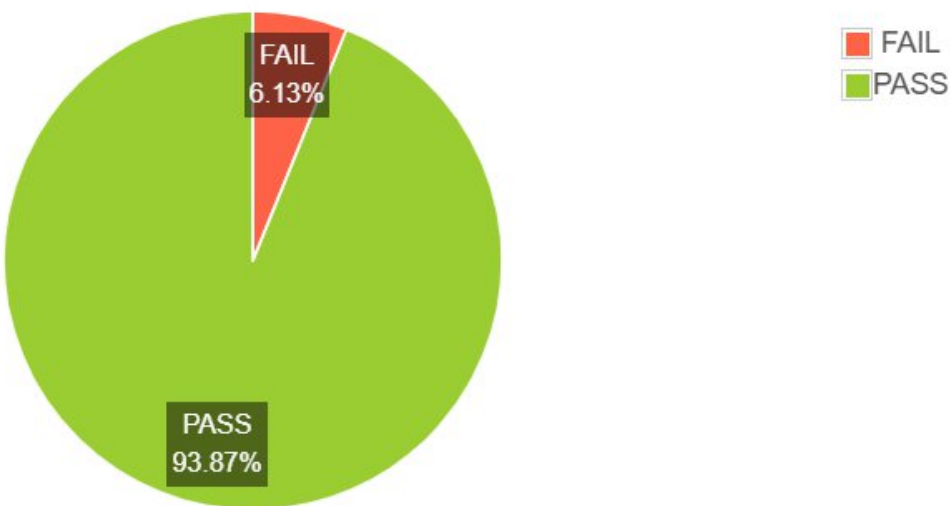
Startup delay (seconds):

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

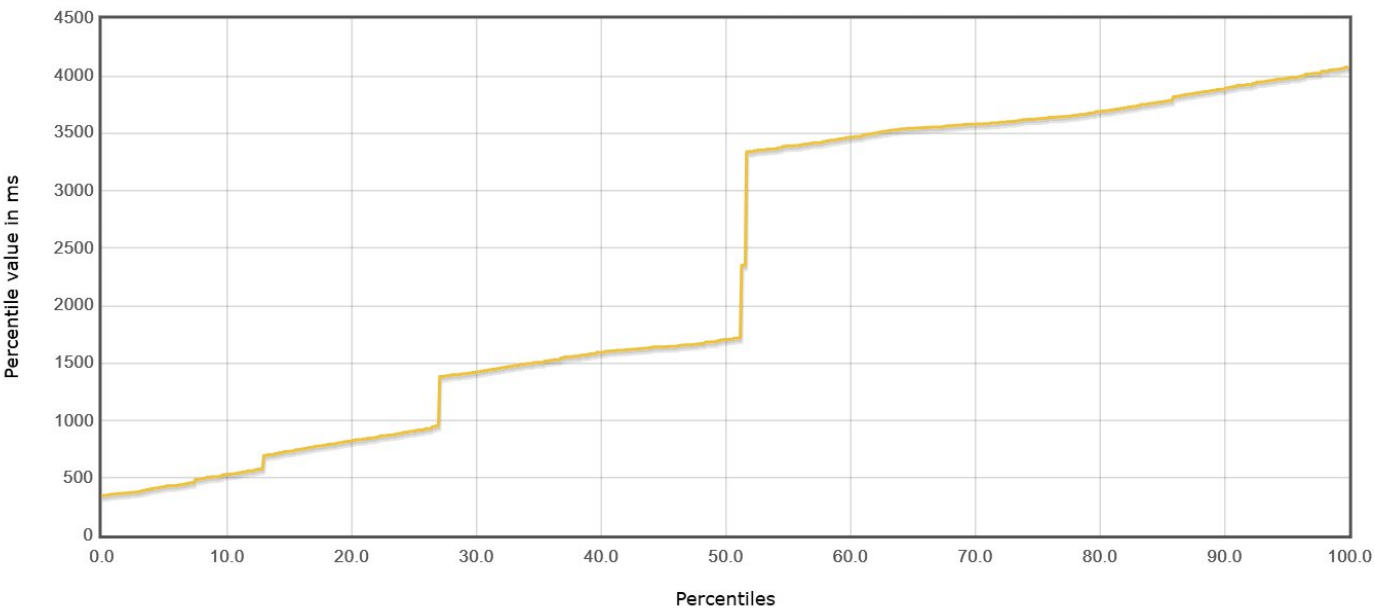
310 пользователей

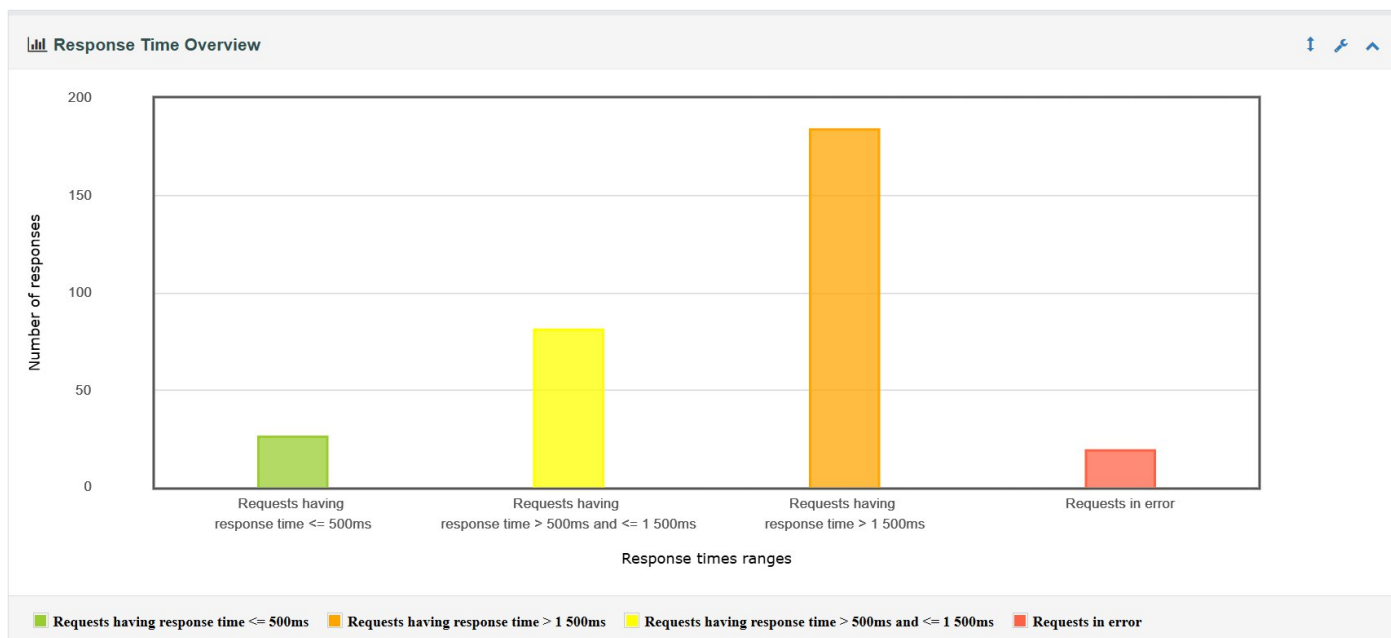


Requests Summary

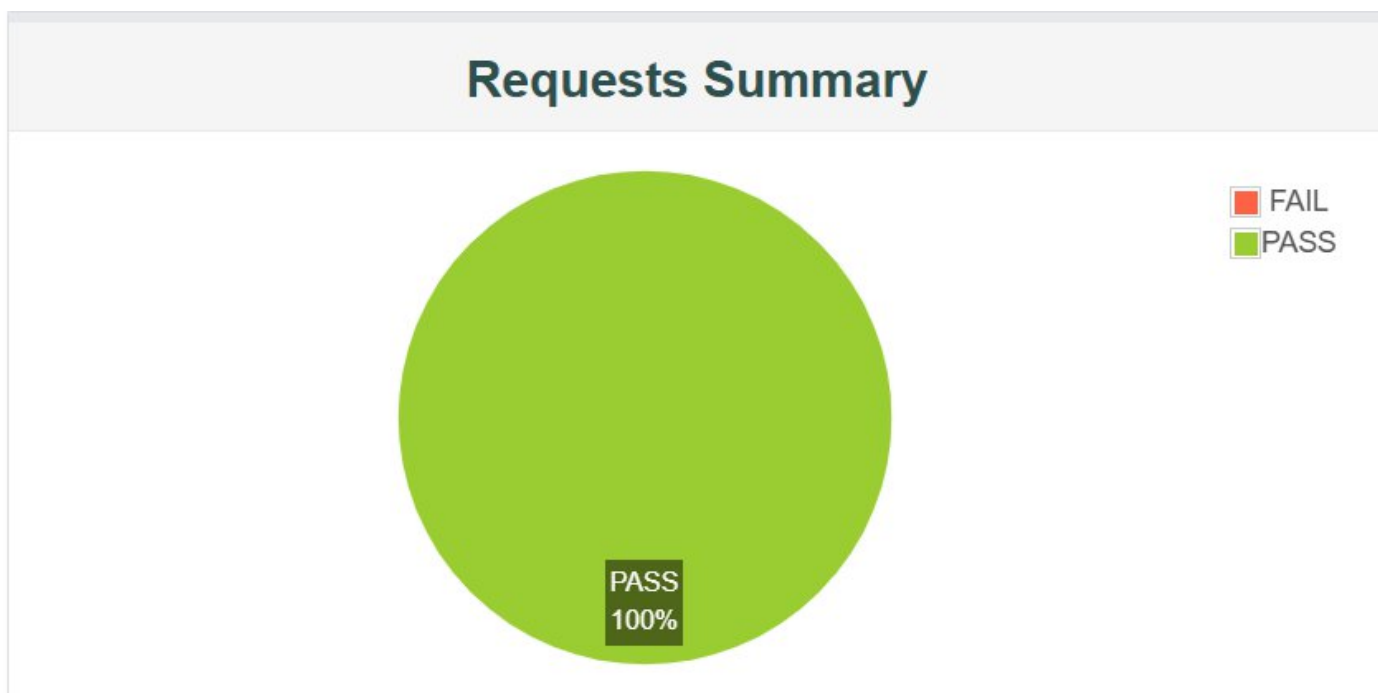


Response Time Percentiles

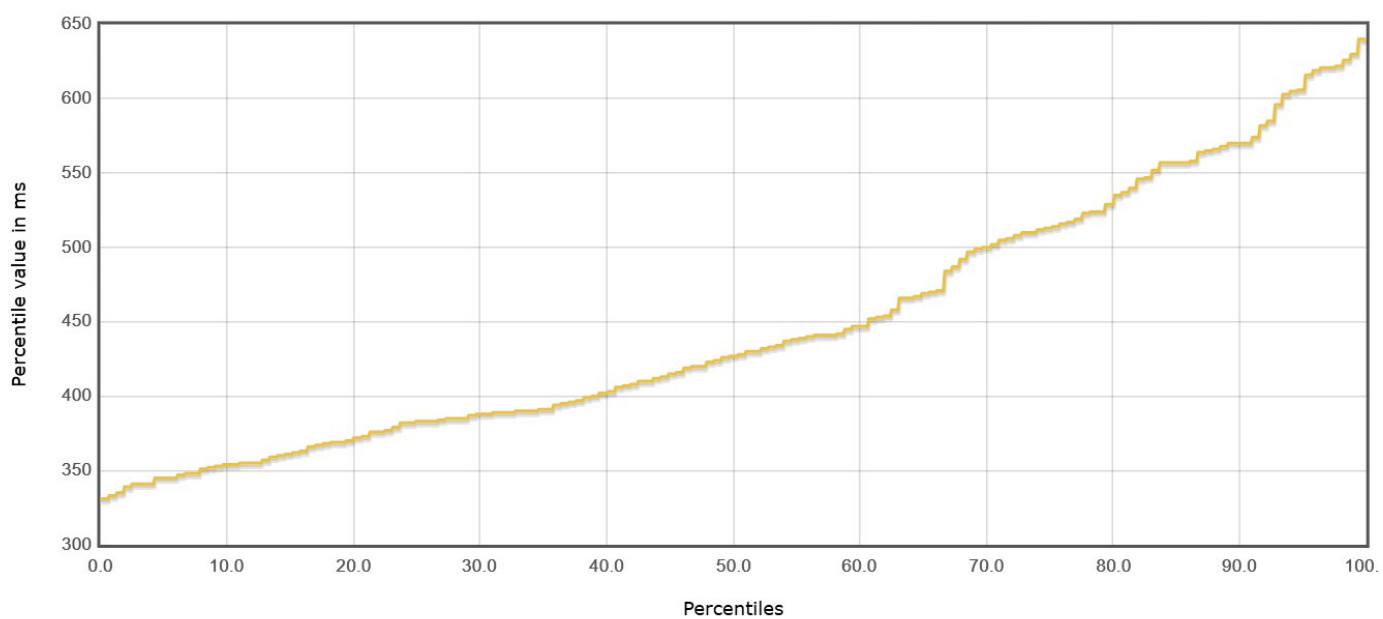




Видим, что при 310 пользователях мы получаем ошибку 503. Попробуем теперь уменьшить количество юзеров, чтобы сайт успевал обрабатывать нагрузку. Возьмём 300 пользователей.



Response Time Percentiles



■ HTTP Request

Теперь видим, что сайт справляется с нагрузкой и мы укладываемся в диапазон 640 мс времени отклика.

Вывод

Выполнив данную лабораторную работу, я на практике познакомился с нагрузочным и стресс тестированием. Вспомнил как правильно конфигурировать программу JMeter для правильного тестирования. Также выяснил какая конфигурация больше подходит к моему заданию.