Министерство образования и науки Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО»
Институт компьютерных наук и кибербезопасности
Высшая школа технологий искусственного интеллекта
Направление: 02.03.01 Математика и компьютерные науки

Отчёт по дисциплине «Программирование на языке JAVA»
Лабораторная работа №3
«Load testing»

Студент: _____ Богдан А. В.
группы 5130201/20002
Преподаватель: _____ Лукашин А. А.

«_____»_____ 2025г.

Санкт-Петербург, 2025

1 Постановка задачи

- Create a separate project to measure performance of your HTTP server and JSON parser
- You can use load testing frameworks (like JMeter for example) or write your own scripts
- HTTP server from lab-2 should be configured to:
  - Request 1: Accept request, parse JSON, store something in file (or in database, you could use something like SQLite, but NOT in-memory DB), retrieve something from file
  - Request 2: Accept request, parse JSON, get something from memory (or perform calculations), create and return JSON
- Combine different variants: Virtual/Classic Threads, your JSON library (lab-1) / Jackson or Gson
- It will be good to run load tests on separate machine (one for HTTP server and another for tests)
- The report must contain:
  - How to configure and launch (README)
  - Experiment description
  - Hardware description
  - Experiment parameters (number of threads, number of requests, amount of data etc)
  - resulting table
- The table with results must contains:

req	Virtual + own parser	Virtual + GSON	Classic + own parser	Classic + GSON
Request-1	avg time per request	...	...	...
Request-2	avg time per request	...	...	...

2 Описание эксперимента

Эксперимент проводился на одной вычислительной машине, на ней запускался и JMeter и сервер.

2.1 Цель

Оценить производительность HTTP-сервера при различных конфигурациях:

- Виртуальные или классические потоки.
- Собственный парсер или GSON.

2.2 Описание аппаратного обеспечения

Серверная часть:

- Тип сервера: Физический сервер.
- Процессор: Intel Core Ultra 7 155H, 16 ядер, 22 потока 3.8-4.8 GHz.
- Оперативная память: 16 ГБ LPDDR5.
- Хранилище: SSD NVMe, 1 ТБ.
- Хранилище: Операционная система: Windows 11 Домашняя.

2.3 Описание параметров эксперимента

Для проведения нагрузочного тестирования была использована программа Apache Jmeter. Для всех восьми экспериментов были выставлены следующие параметры тестового плана:

- Количество потоков (пользователей): 100.
- Время, в течение которого будут прибавляться пользователи: 1.
- Количество раз выполнения тестирования: 10.
- Количество запросов: 10000.
- Объём данных: JSON-запрос для 1 запроса - 91 байт, JSON-запрос для 2 запроса - 55 байт. JSON-ответ для 1 запроса - 9 байт, JSON-ответ для 2 запроса - 4.75 Кбайта.

На рисунке 1 показаны общие параметры запросов - хост и порт, а на рисунке 2 и 3 отображены параметры и тела запросов 1 и 2.

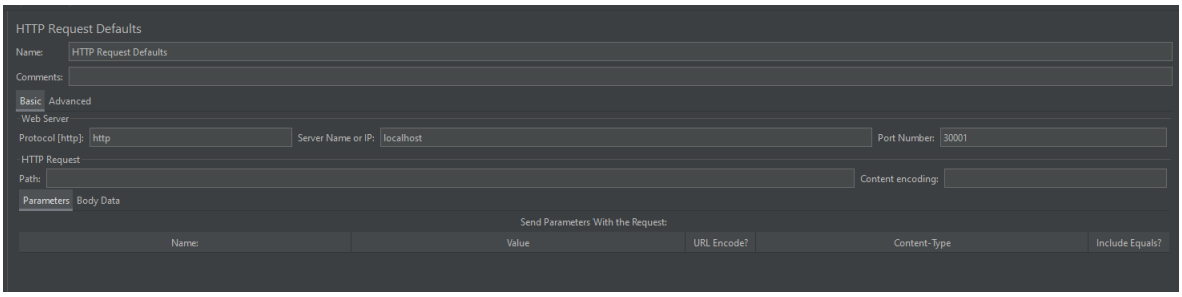

Рис. 1. Общие параметры запросов.

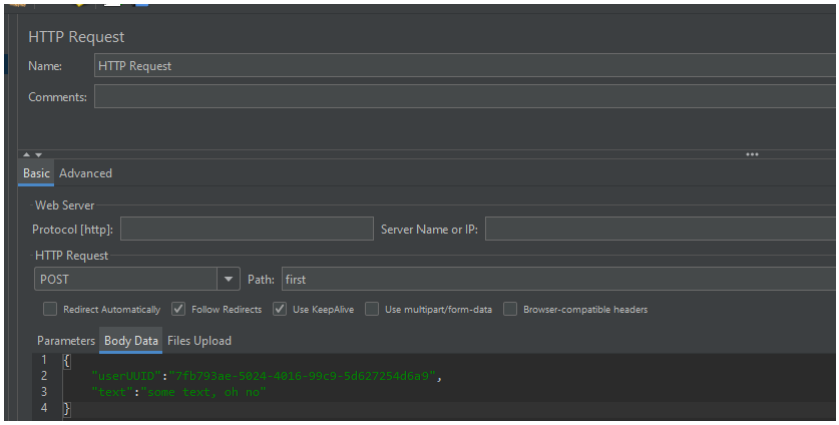

Рис. 2. Параметры первого запроса.

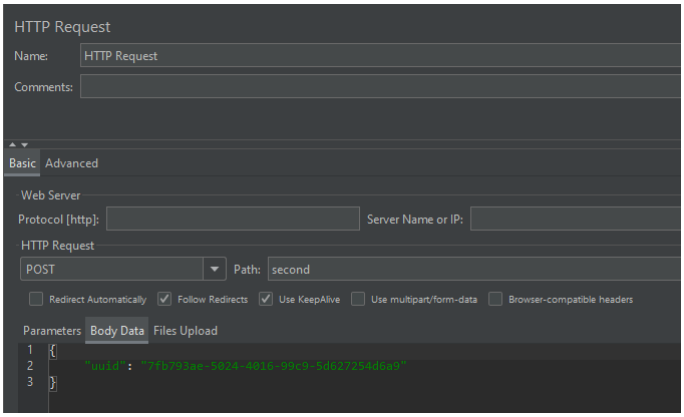

Рис. 3. Параметры второго запроса.

3 Результаты

Все результаты проведения нагрузочного тестирования представлены на рисунках 4-11.

Рис. 4 Результаты выполнения 1 запроса с Virtual + own parser конфигурацией.

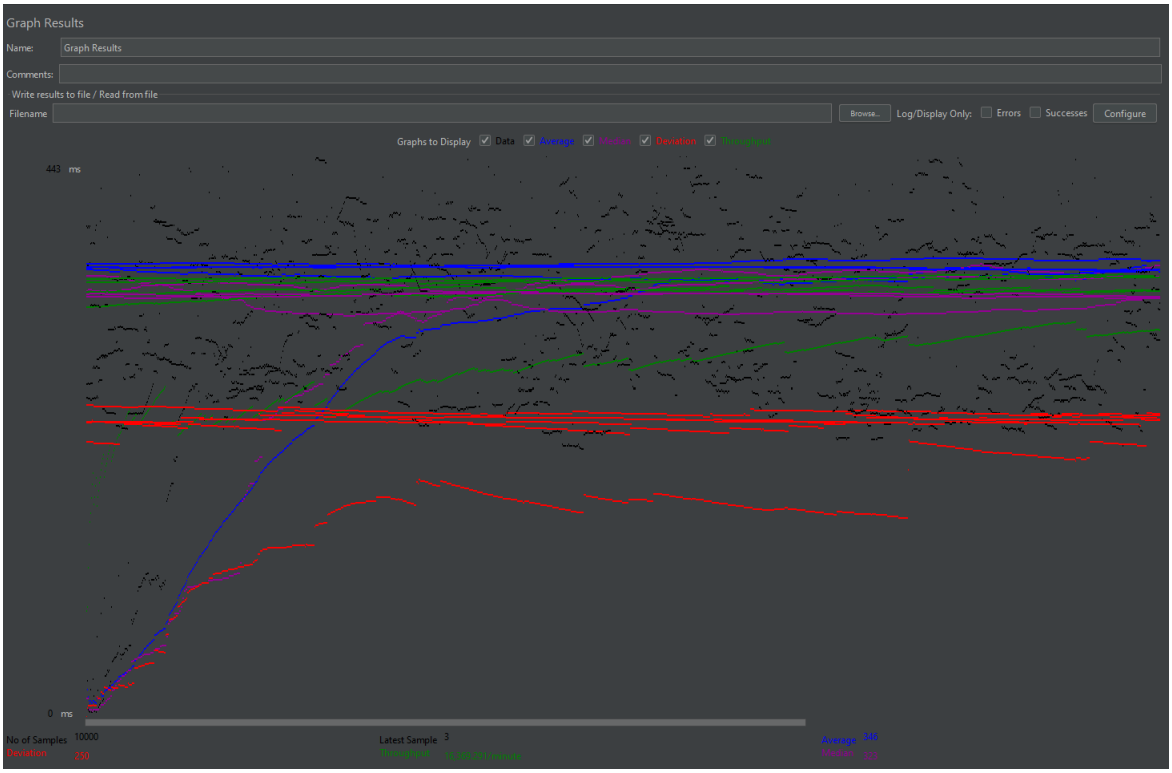

Рис. 5. Результаты выполнения 1 запроса с Virtual + GSON конфигурацией.

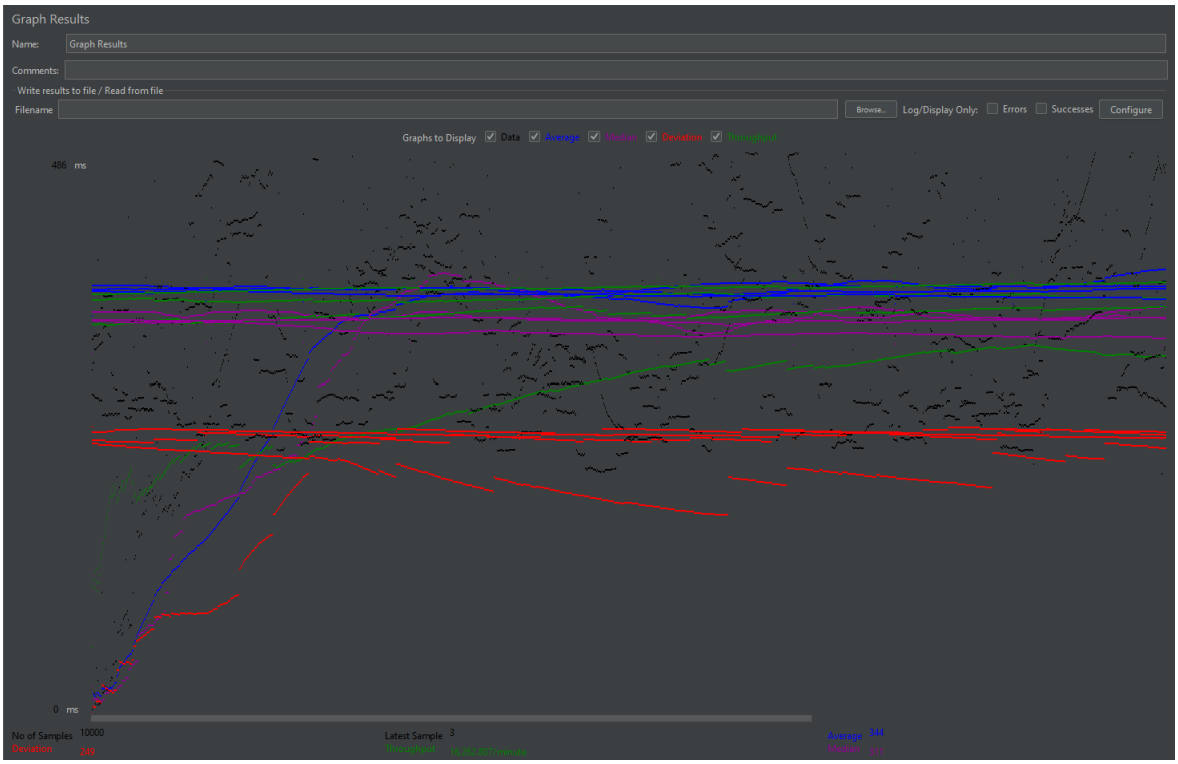

Рис. 6. Результаты выполнения 1 запроса с Classic + own parser конфигурацией.

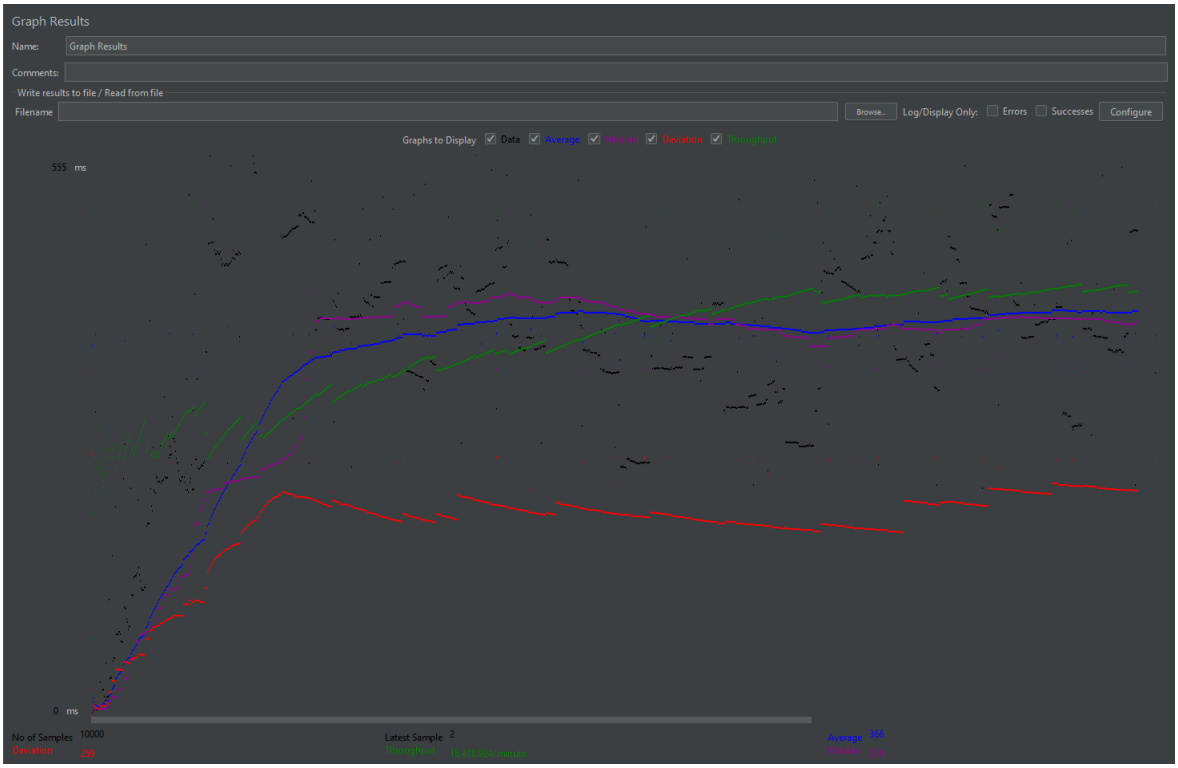

Рис. 7. Результаты выполнения 1 запроса с Classic + GSON конфигурацией.

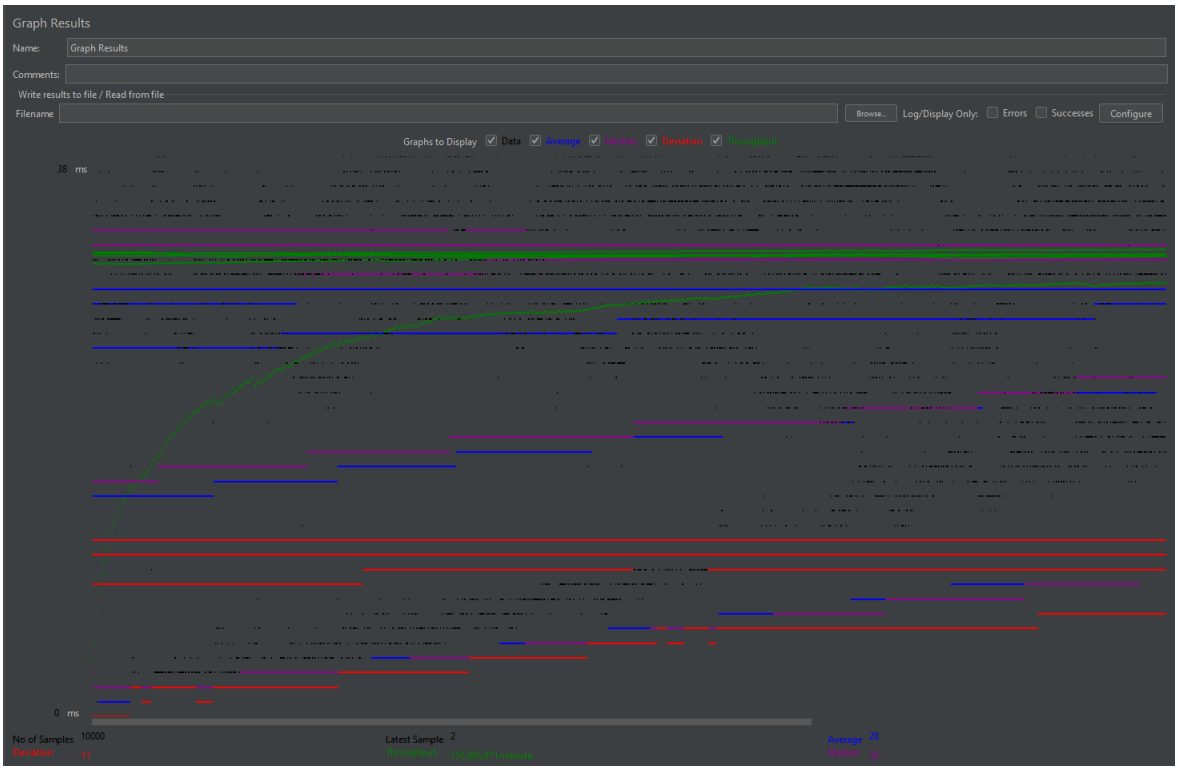

Рис. 8. Результаты выполнения 2 запроса с Virtual + own parser конфигурацией.

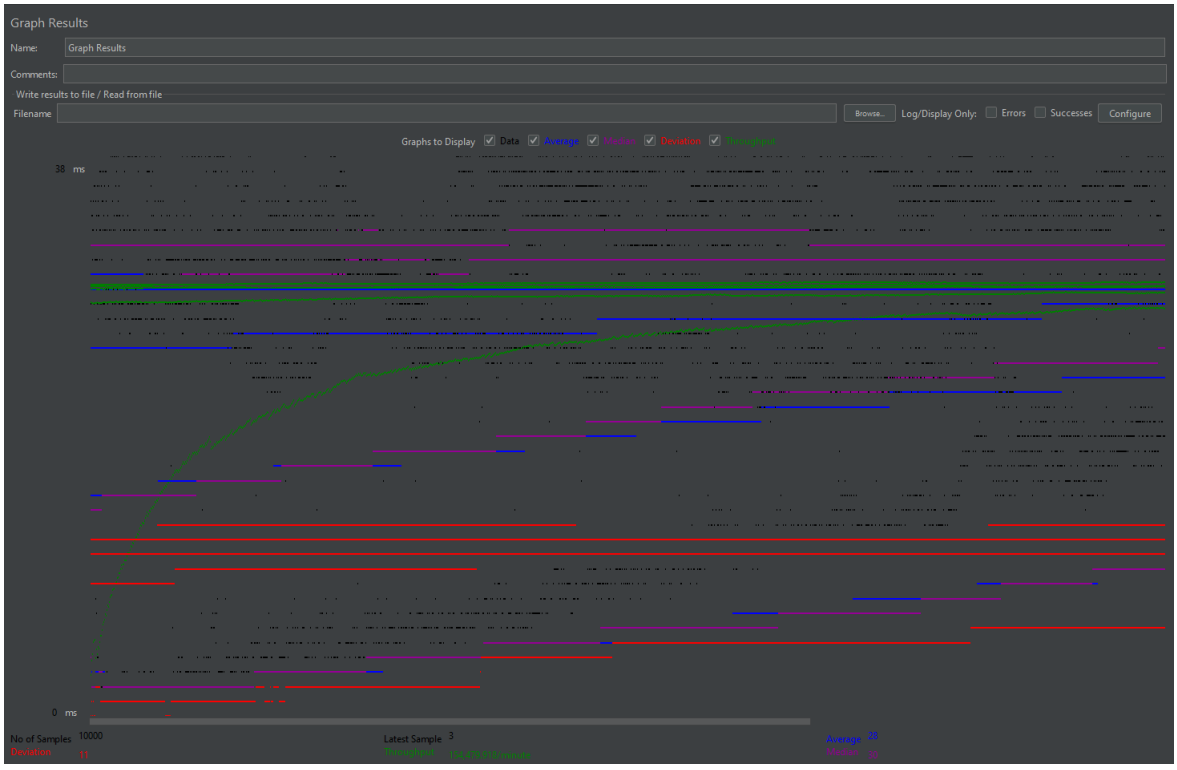

Рис. 9. Результаты выполнения 2 запроса с Virtual + GSON конфигурацией.

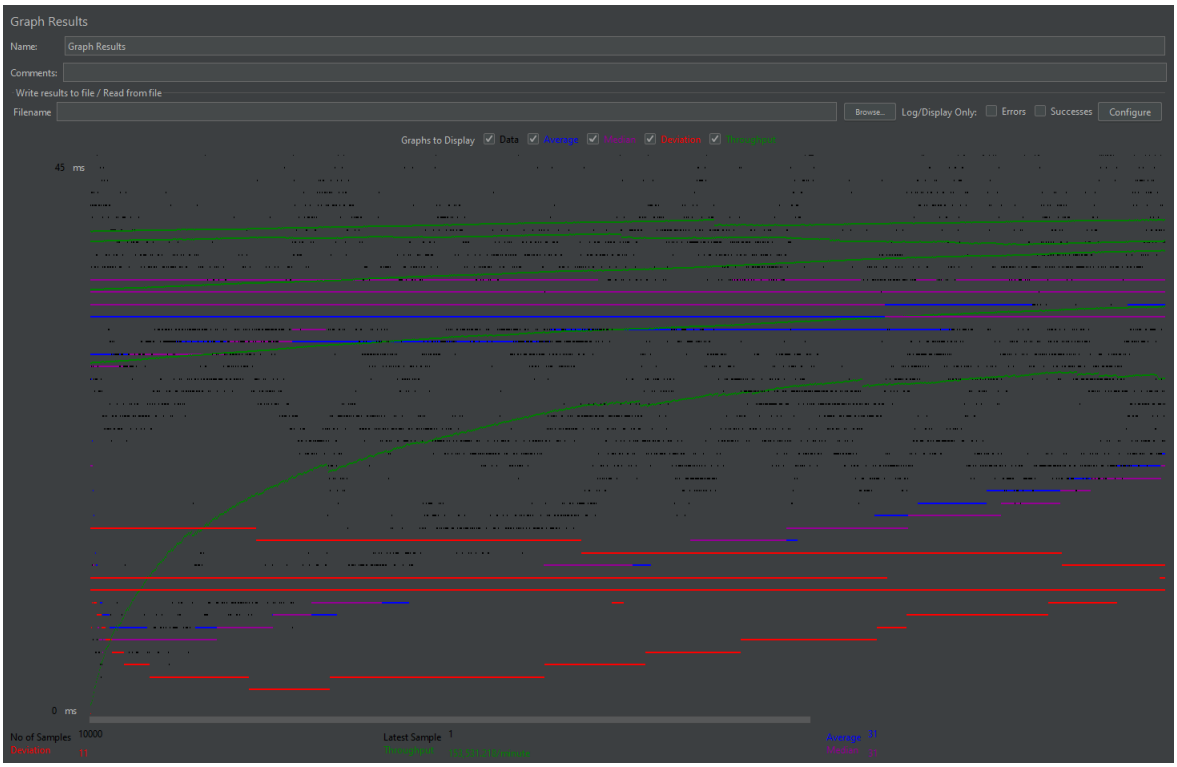

Рис. 10. Результаты выполнения 2 запроса с Classic + own parser конфигурацией.

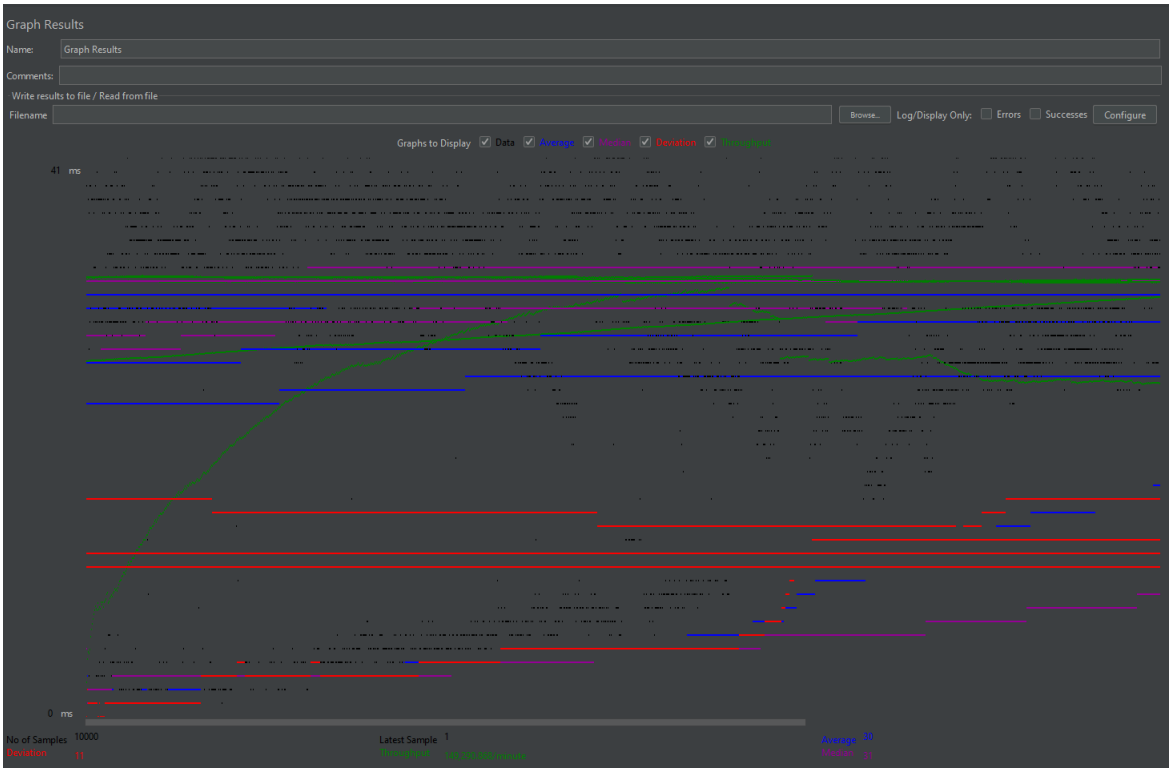

Рис. 11. Результаты выполнения 2 запроса с Classic + GSON конфигурацией.

В таблице 1 представлены результаты эксперимента.

Таблица 1. Результаты эксперимента

req	Virtual + own parser	Virtual + GSON	Classic + own parser	Classic + GSON
Request-1	3.74 мс	3.66 мс	3.69	3.89 мс
Request-2	0.382 мс	0.388 мс	0.391 мс	0.402 мс

На основе проведённого эксперимента можно сделать следующие выводы:

1. Влияние типа потоков на производительность.
    - Виртуальные потоки показали сравнимую производительность с классическими потоками для обоих типов запросов.
    - Для Request-1 время обработки с виртуальными потоками составило 3.74 мс (с собственным парсером) и 3.66 мс (с GSON).
    - Для Request-2 время обработки с виртуальными потоками составило 0.382 мс (с собственным парсером) и 0.388 мс (с GSON).
    - Виртуальные потоки демонстрируют незначительное преимущество в скорости обработки, что может быть связано с их оптимизацией для задач с высокой конкуренцией за ресурсы.
2. Влияние парсера на производительность.
    - Собственный парсер показал немного лучшую производительность в 3 из 4 случаев по сравнению с GSON.
    - Для Request-1 время обработки с собственным парсером составило 3.74 мс (виртуальные потоки) и 3.69 мс (классические потоки).
    - Для Request-2 время обработки с собственным парсером составило 0.382 мс (виртуальные потоки) и 0.391 мс (классические потоки).
    - Собственный парсер демонстрирует немного лучшую производительность, чем GSON, однако разница незначительна, что говорит о высокой оптимизации GSON.

4 Заключение

Проведённый эксперимент показал, что:

1. Виртуальные потоки демонстрируют сопоставимую или немного лучшую производительность по сравнению с классическими потоками.
2. Собственный парсер немного быстрее GSON, но разница незначительна.