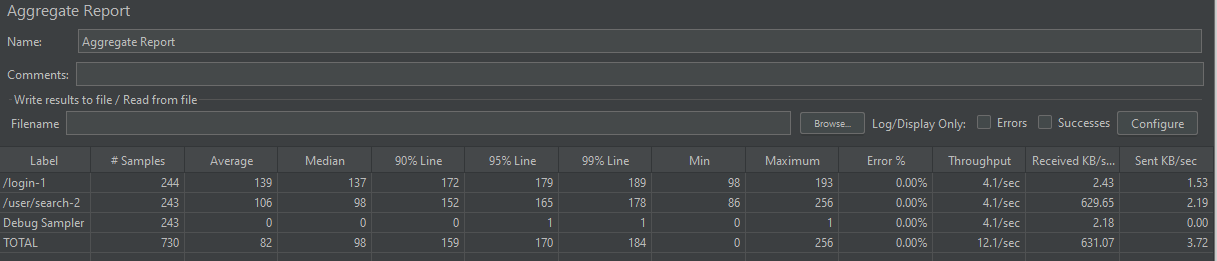
## **Сценарии**

**Дата теста** 22.09.2024

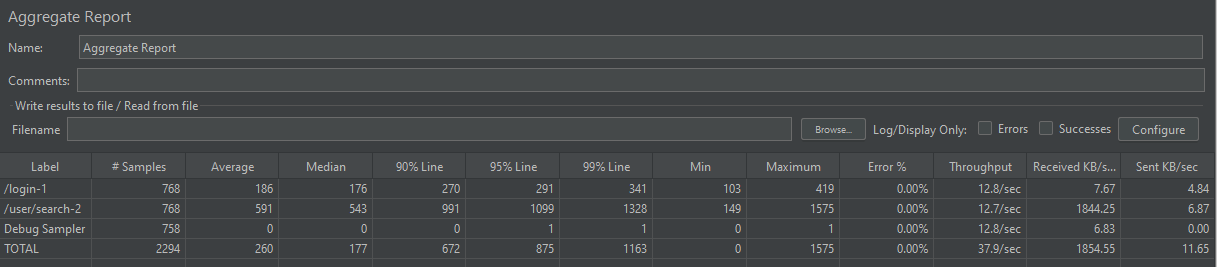
1. Поиск пользователя (эндпойнт /user/search) с количеством одновременных запросов 1/10/100/1000 на протяжении 60 секунд.
2. Создание индекса по столбцам first\_name, second\_name. Поиск пользователя (эндпойнт /user/search) с количеством одновременных запросов 1/10/100/1000 на протяжении 60 секунд.

## **Основные выводы**

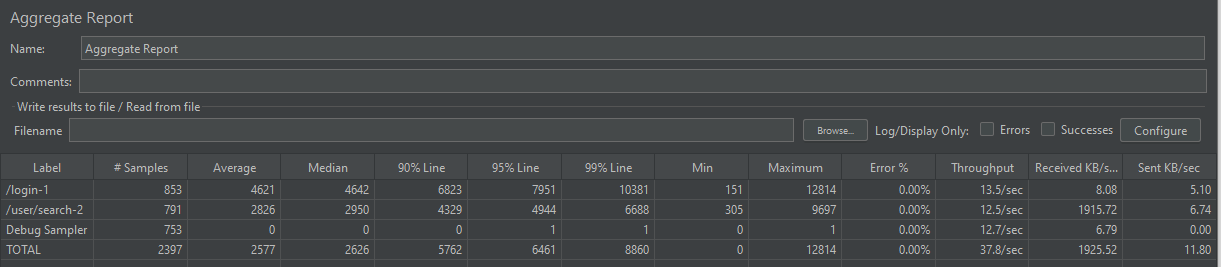
* 1 поток 60 сек



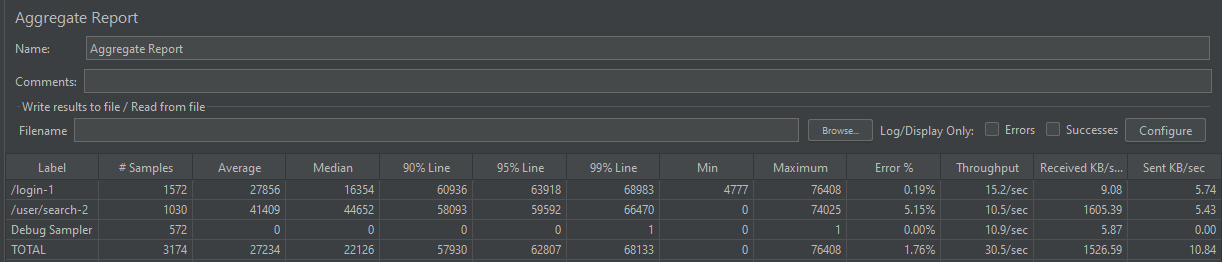
* 10 потоков 60 сек



* 100 потоков 60 сек

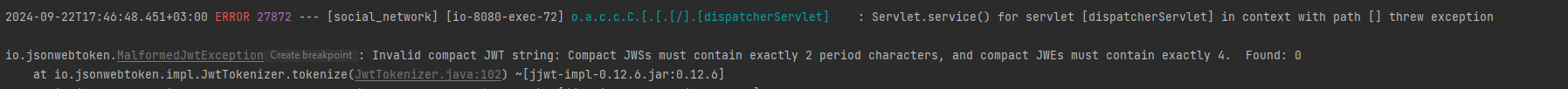


* 1000 потоков 60 сек



По первому этапу можно сделать следующие выводы:

1. По мере увеличения количества потоков 1 -> 10 -> 100 -> 1000 на 90 перцентиле зафиксированы следующие изменения (эндпойнт /user/search) latency: 152 -> 543 -> 2950 -> 44652
2. Изменение throughput: 4.1/sec -> 12.7/sec -> 12.5/sec -> 10.5/sec
3. При 1000 потоках в количестве запросов появились ошибки в получении токена авторизации `MalformedJwtException`.



Данные ошибки можно связать с:

* Проблемы с сетью:

- Высокая нагрузка может вызвать задержки или потерю данных в сети, что может привести к неполучению полного JWT или его повреждению.

* Ограничения ресурсов:

- Сервер может исчерпать ресурсы (например, память или процессорное время), что может привести к некорректной обработке JWT.

* Ошибки в коде обработки JWT:

- При высокой нагрузке могут проявляться ошибки в коде, которые не были очевидны при меньшей нагрузке, такие как неправильное управление потоками или некорректная обработка исключений.

На втором этапе создаем индекс:

**CREATE** **INDEX** idx\_users\_first\_second\_name\_pattern **ON** public.users **USING** btree (first\_name text\_pattern\_ops, second\_name text\_pattern\_ops);

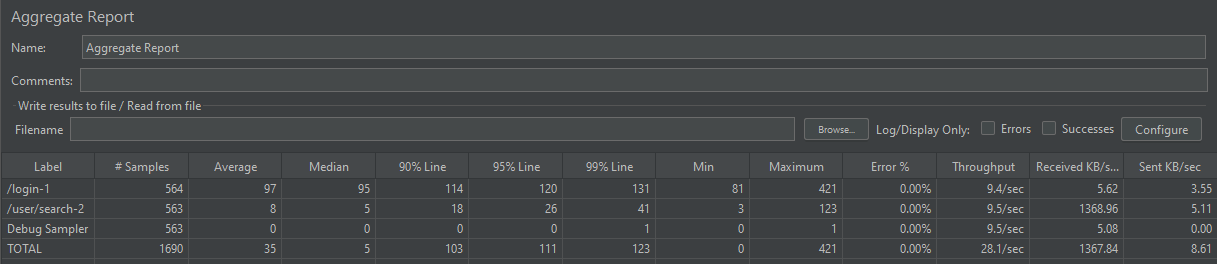
При создании индекса используем текстовый паттерн text\_pattern\_ops.

Операторный класс `text\_pattern\_ops` оптимизирует индексацию для операций, которые используют шаблоны поиска, такие как `LIKE 'prefix%'`. Это позволяет избежать полного сканирования таблицы и использовать индекс для быстрого поиска строк, начинающихся с определенного префикса.

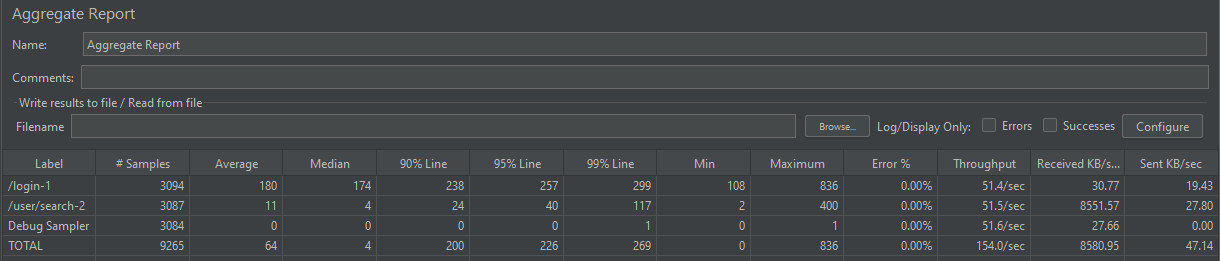
Однако, `text\_pattern\_ops` не поддерживает поиск с использованием шаблонов, которые начинаются с `%`, например, `LIKE '%suffix'`

**Создан индекс**

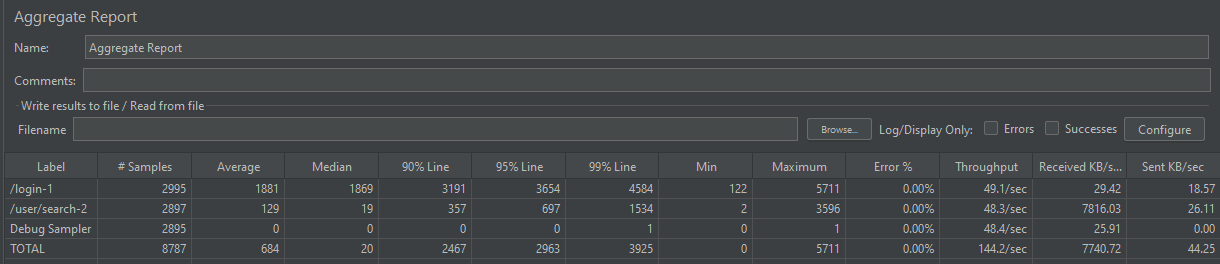
* 1 поток 60 сек



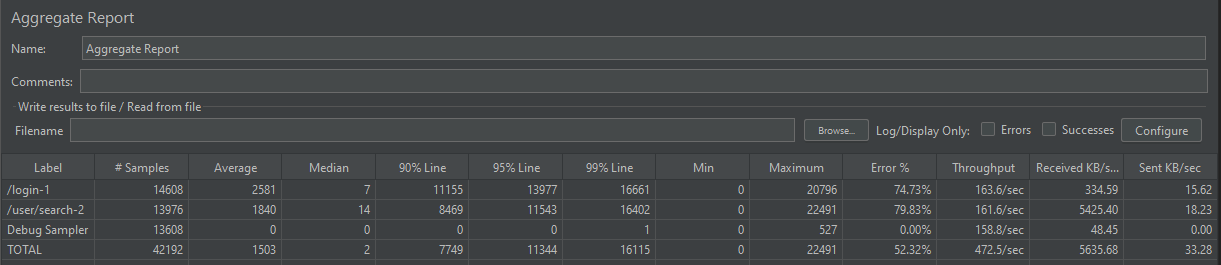
* 10 потоков 60 сек



* 100 потоков 60 сек



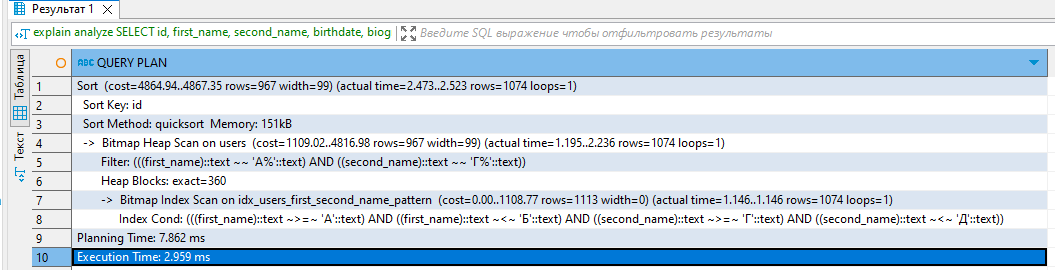
* 1000 потоков 60 сек



По второму этапу можно сделать следующие выводы:

1. По мере увеличения количества потоков 1 -> 10 -> 100 -> 1000 на 90 перцентиле зафиксированы следующие изменения (эндпойнт /user/search) latency: 18 -> 24 -> 357 -> 8469. Отмечаем многократное (до 10 раз) увеличение скорости обработки запроса после добавления индекса.
2. Изменение throughput: 9.5/sec -> 51.5/sec -> 48.3/sec -> 161.6/sec. Отмечаем многократное увеличение объёма передачи информации после добавления индекса.
3. Однако при 1000 потоках в количестве запросов ошибки в получении токена авторизации `MalformedJwtException` сохранились.

**Explain Analyze запроса:**



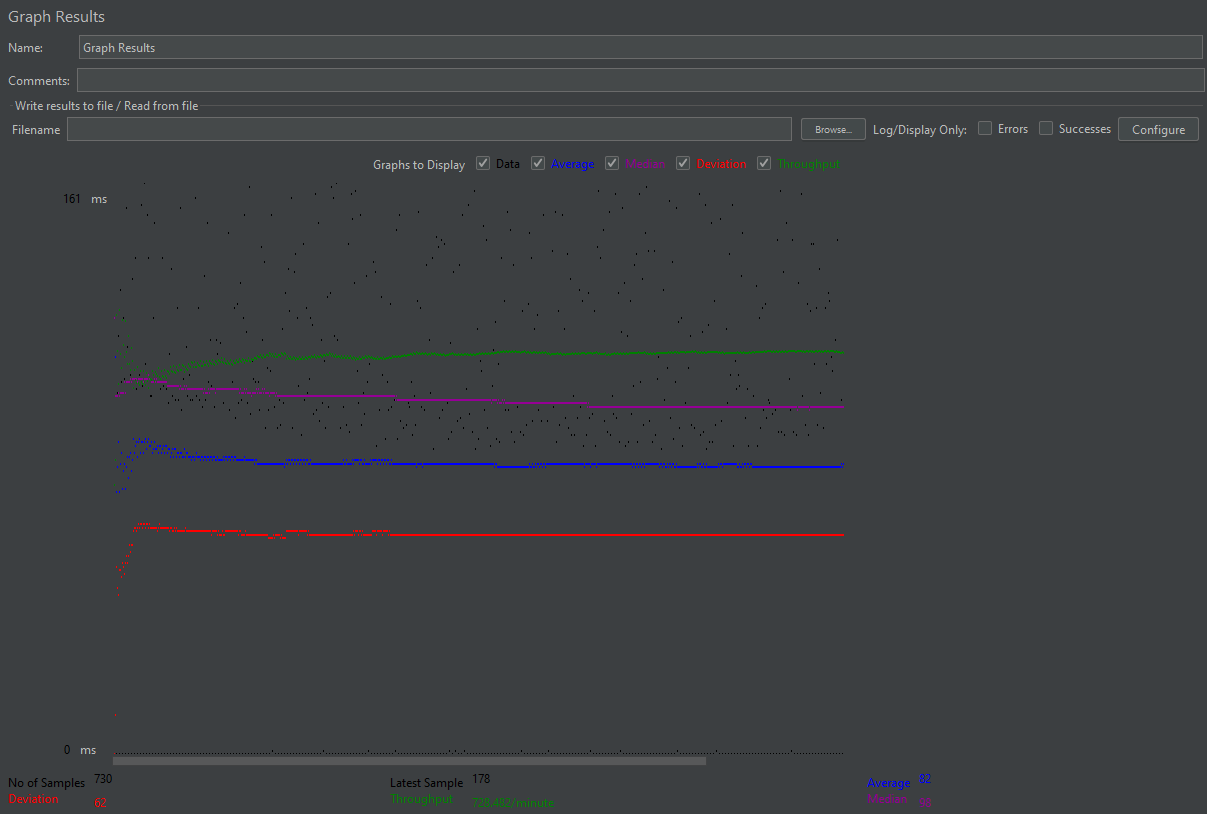
Из анализа запроса видим использование созданного индекса для обработки информации.

## **Рекомендации**

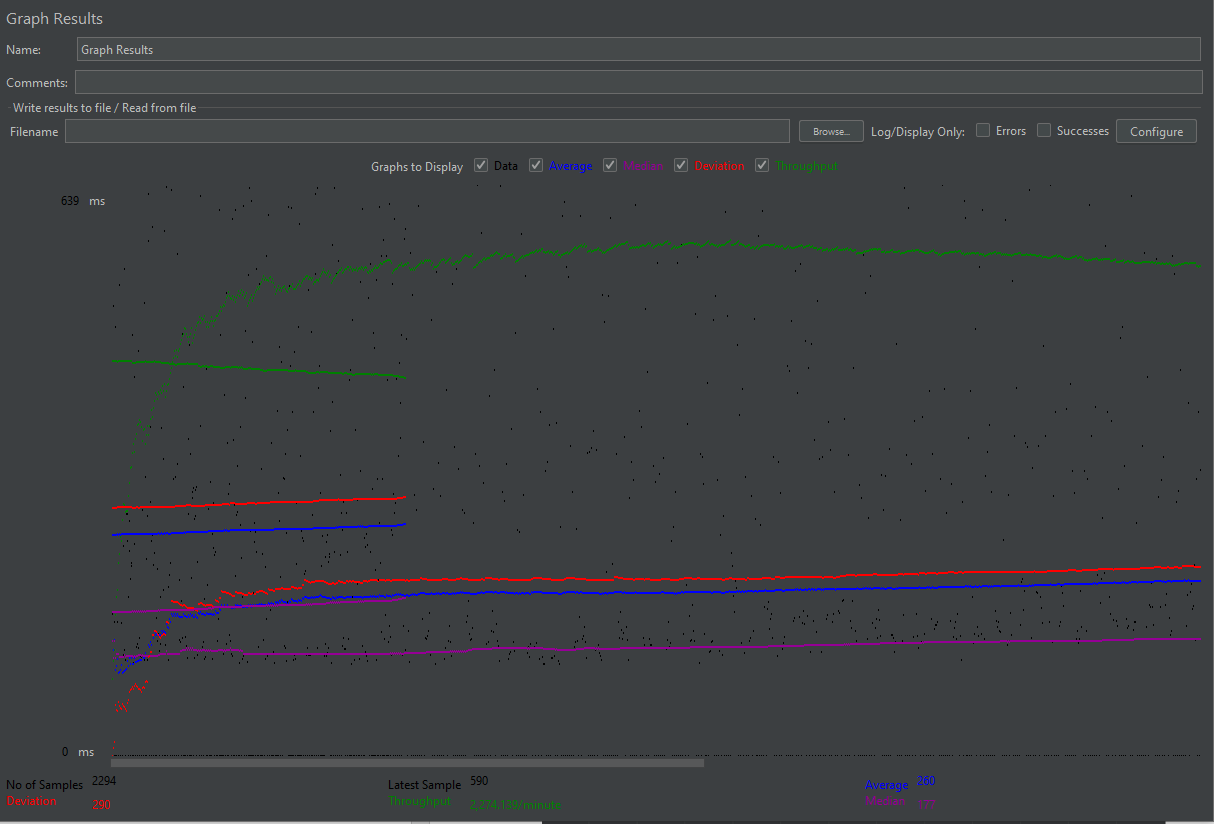
* Обратить внимание на работу метода, генерирующего токены. По возможности провести оптимизацию.

## **Результаты**

1 поток 60 сек



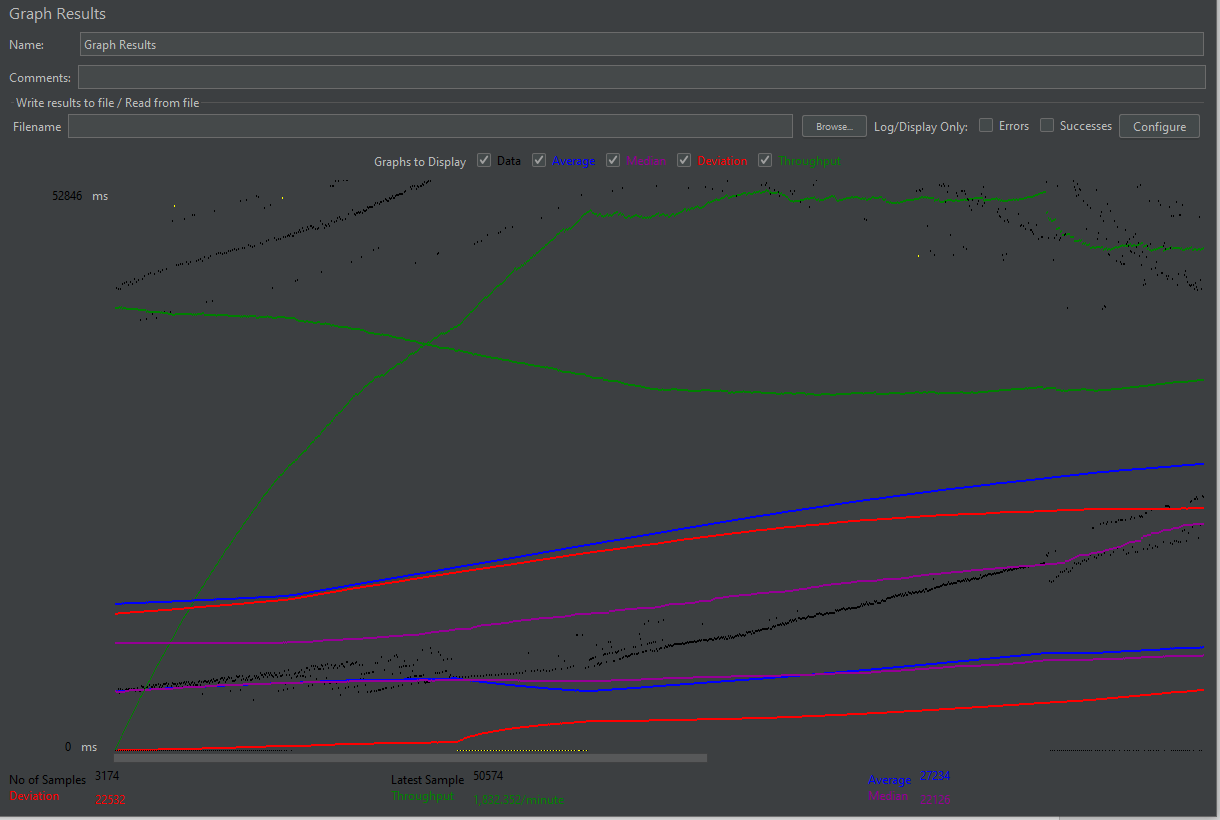
10 потоков 60 сек



100 потоков 60 сек

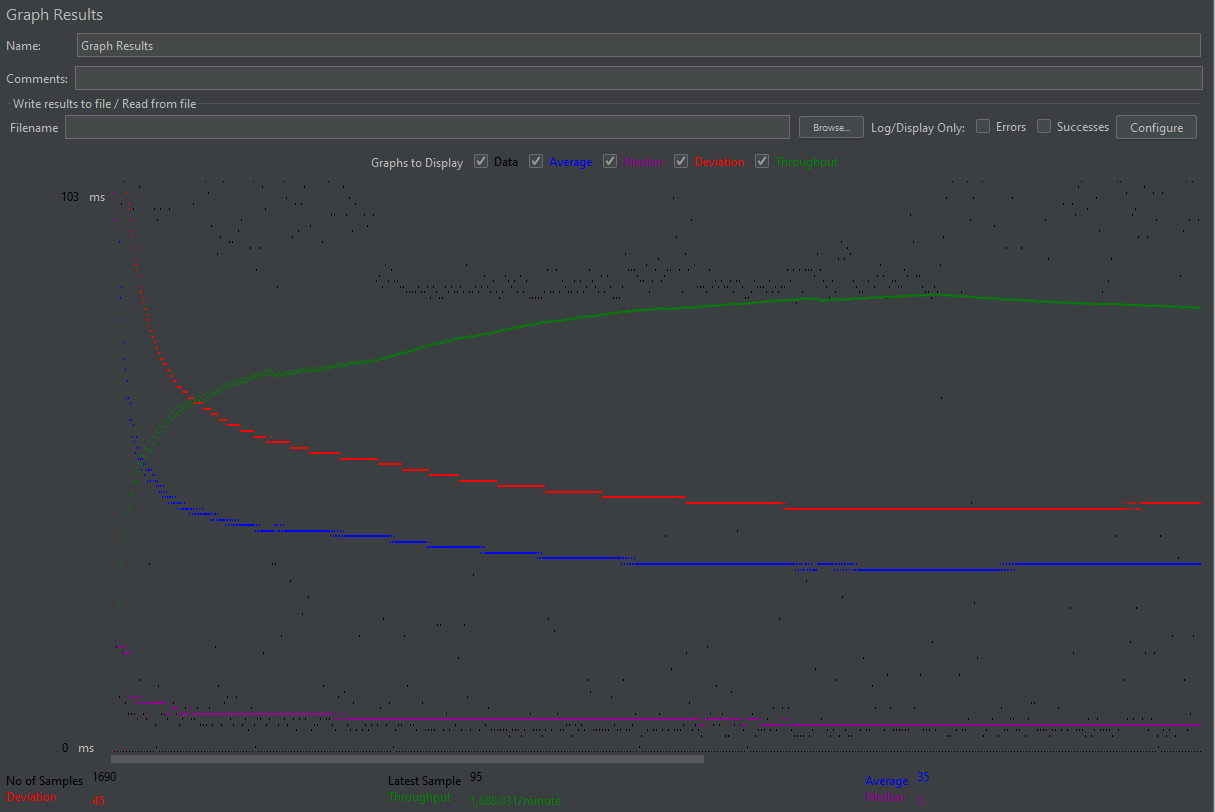


1000 потоков 60 сек

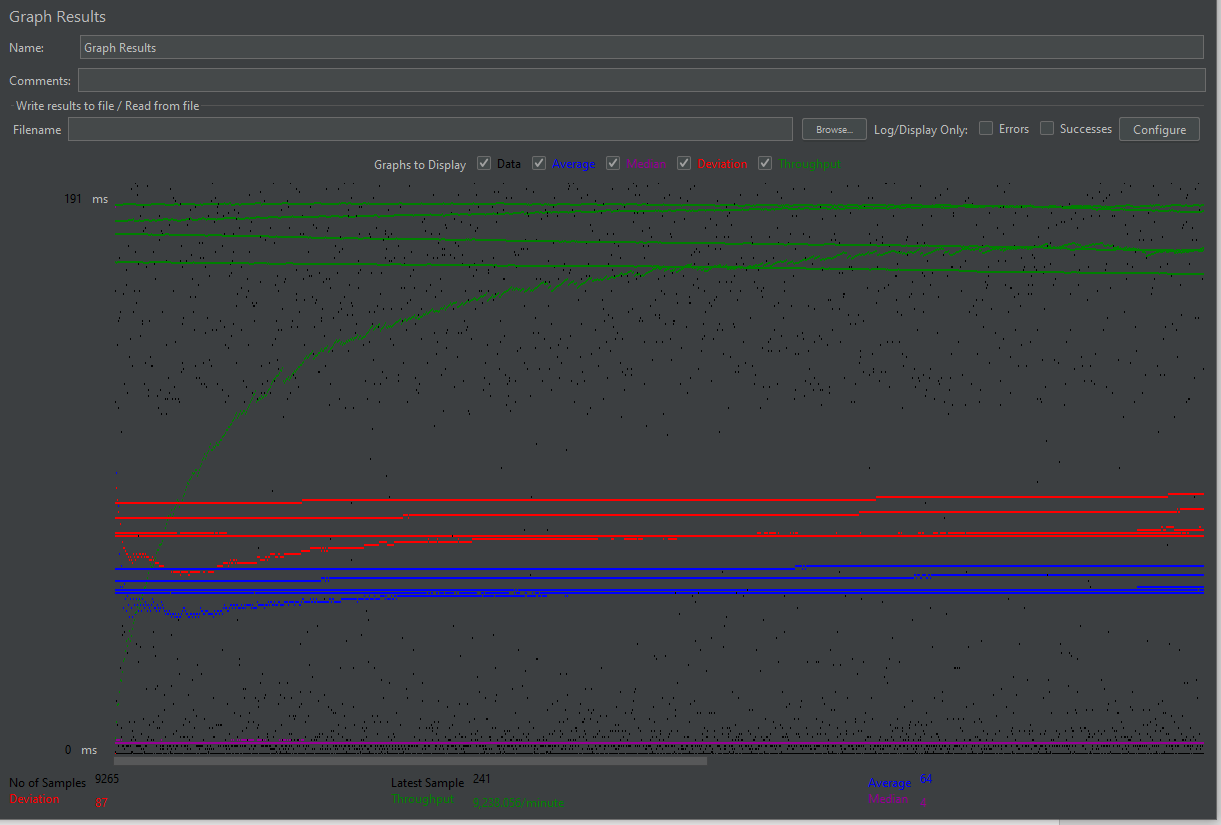


**Создан индекс**

1 поток 60 сек



10 потоков 60 сек



100 потоков 60 сек



1000 потоков 60 сек

