# Отчет о нагрузочном тестирование для курса Highload Architect,

# урок 5

## Описание среди на которой выполняются тесты:

1. Параметры хост машины, на котором запущен docker-compose с базой данных Postgres:



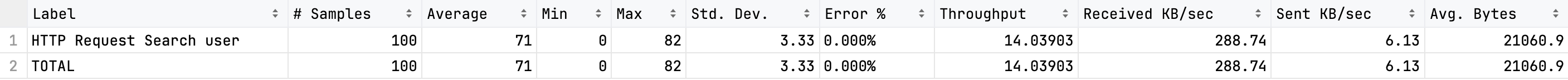
1. Параметры среды, выделенные для окружения docker

Для docker среды выделены значения по умолчанию 1 процессорное ядро, 2 ГБ оперативной памяти и 10 ГБ на SSD диске.

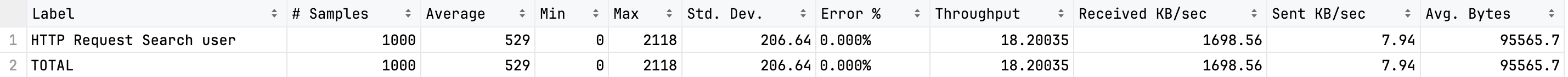
1. Выполняется нагрузка на метод /user/search из спецификации задачи по курсу (поиск анкеты по фамилии и имени, одновременное выполнение условий что имя и фамилия начинаются со значений в переданных параметрах)

## Тесты без индекса, результаты исполнения:

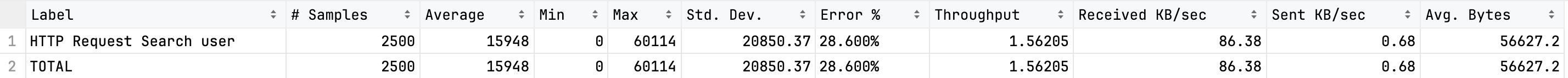
1. 1 поток, 100 итераций:



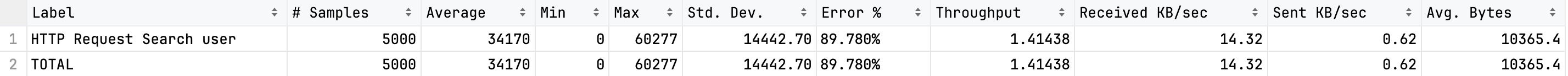
1. 10 потоков, 100 итераций:



1. 25 потоков 100 итераций



1. 50 потоков, 100 итераций:

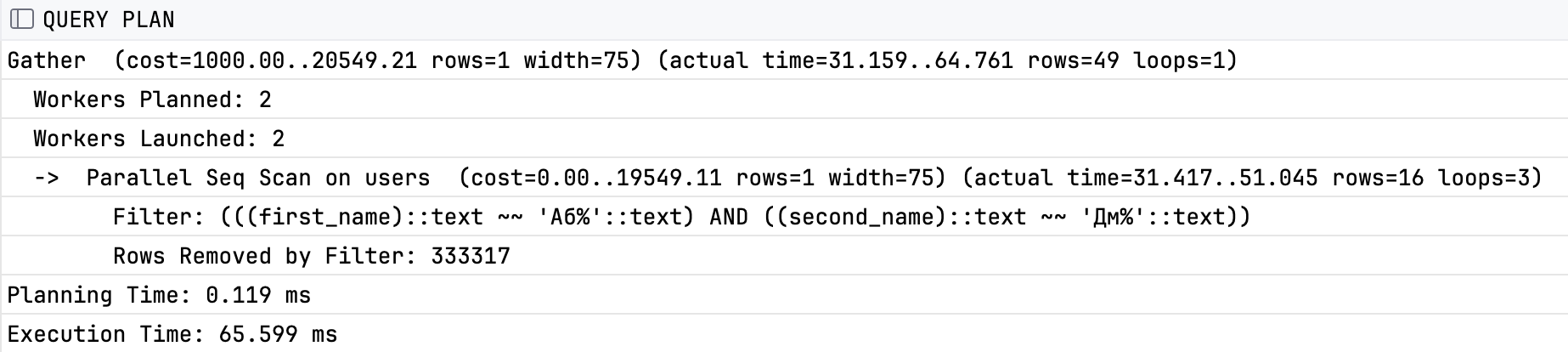


**Выводы:** при отсутствии индекса, уже при нагрузке в 25 потоков приложение испытывает сложности в обработке запросов и почти 30% запросов падают с ошибкой (запрос в базу данных не успевает отработать за 30 секунд)

## Набор действий по созданию оптимизации базы:

1. Анализ плана выполнения запроса без индекса

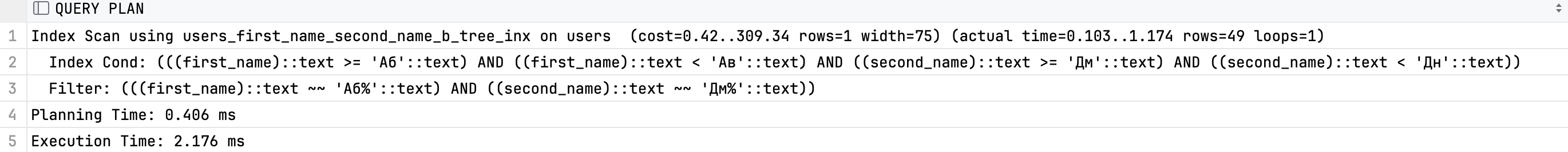
**explain analyse  
select** *\****from** users  
**where first\_name like 'Аб%'  
 and second\_name like 'Дм%'**;



1. Создание индекса

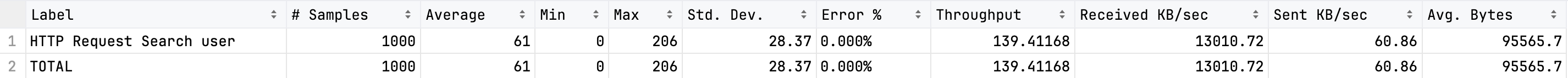
**create index** users\_first\_name\_second\_name\_b\_tree\_inx **on** users (**first\_name collate** "C", **second\_name collate** "C");

1. Анализ плана выполнения запроса с индексом

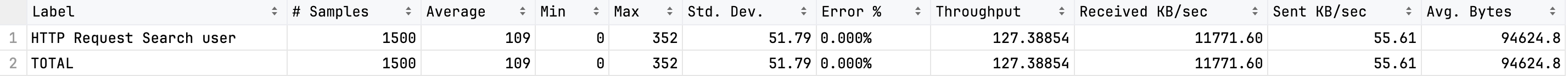


## Тесты с индексом B-tree

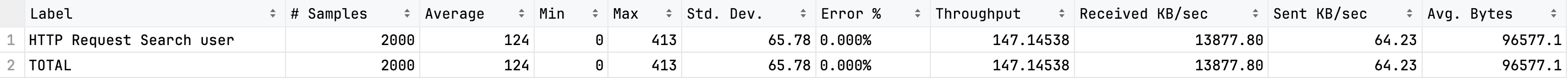
1. 10 потоков, 100 итераций:



1. 15 потоков, 100 итераций:



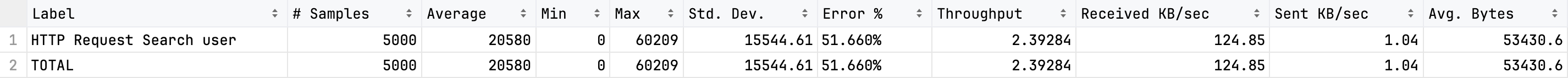
1. 20 потоков, 100 итераций:



1. 25 потоков, 100 итераций:



1. 50 потоков, 100 итераций:



## Выводы

1. Добавление B-tree индекса повышает пропускную способность приложения. Ошибки в ответах появляются при 25 потоках, но их количество меньше 5%. При это даже при 50 потоках, процент ответов с ошибкой около 50%.
2. Пропускная способность так же возросла. Без индекса пропускная способность была на уровне 15-20 запросов в секунду, после добавления индекса, пропускная способность увеличилась до 130-150 запросов в секунду.
3. Снизилось время ответа под нагрузкой, в среднем с 500 миллисекунд почти до 50 миллисекунд.
4. При тестировании не менялись параметры пулов подключения к базе данных и не изменялись параметры самой базы, возможно тут тоже есть возможности для улучшения производительности.