Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования «Кубанский государственный университет»

Кафедра вычислительных технологий

**ОТЧЕТ**

о выполнении лабораторной работы №5

по дисциплине “Информационная безопасность”

Выполнил: ст. гр. 36/2

Баева Д. Н.

Проверил: Усов П. Е.

Краснодар

2023

**ЛВС, web-сервер с CMS. Нагрузочное тестирование web-сервера.**

**Цель работы****:** С помощью систем нагрузочного тестирования определить производительность web-серверов Apache и Nginx, добиться отказа в обслуживании.

**Ход работы**

Часть 1

Для тестирования используются 2 машины – одна с установленным и работающим Apache, вторая будет отсылать запросы и делать выводы о производительности web-сервера.

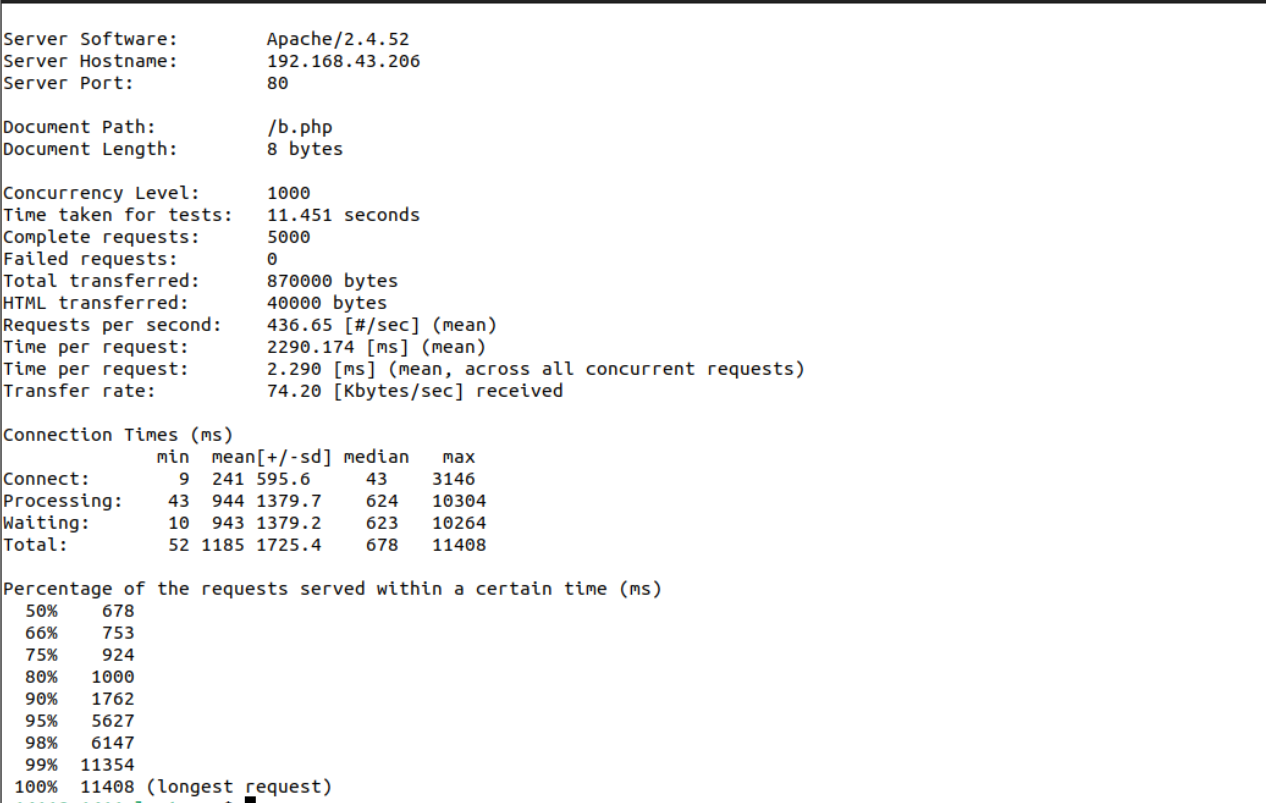


Рисунок 1 – Настройка Apache для первой виртуальной машины.

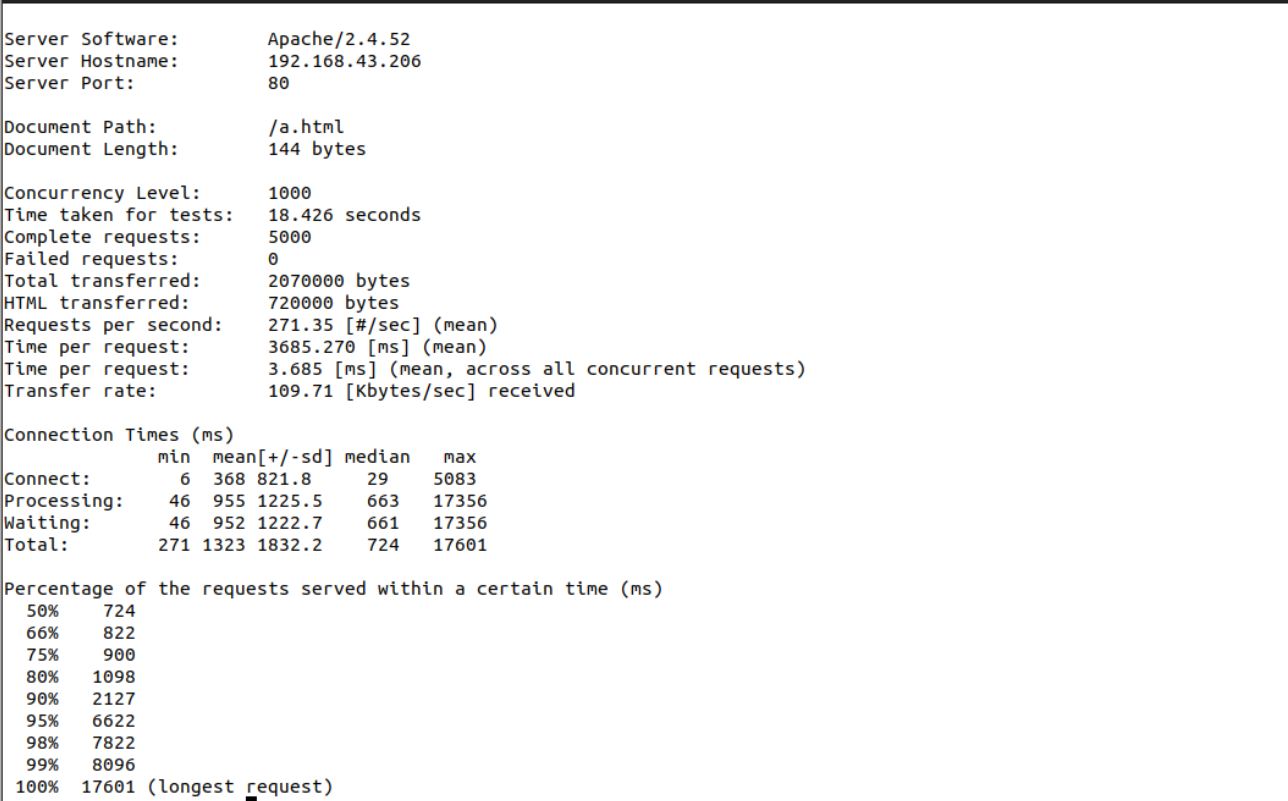


Рисунок 2 – Настройка Apache для второй виртуальной машины.

1. Определить максимальное число параллельных запросов, при котором сервер нас не будет блокировать.

2. Провести тест при использовании максимального числа запросов.

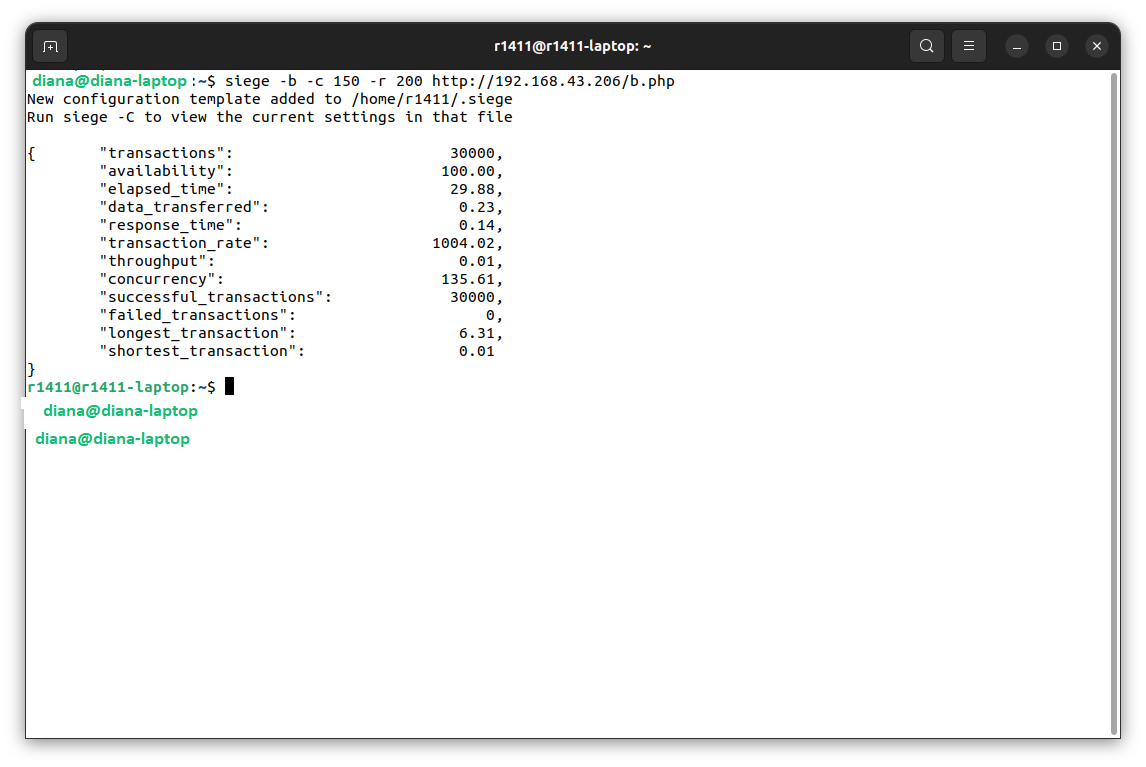


Рисунок 3 – Тестирование на PHP-запросы.

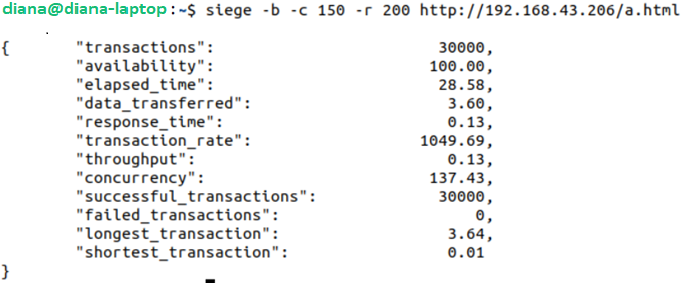


Рисунок 4 – Тестирование на HTML-запросы.

3. Определить максимальное число параллельных запросов.

4. Провести тест при использовании максимального числа запросов.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 5 – Тест при использовании максимального числа запросов (PHP-запросы).

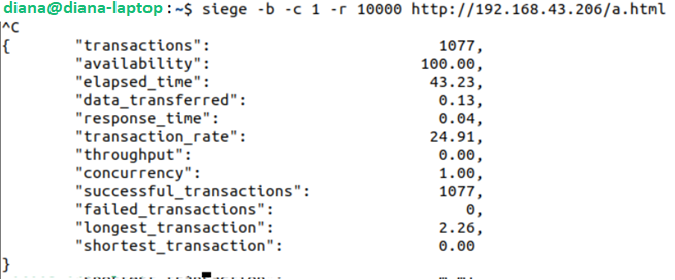


Рисунок 6 – Тест при использовании максимального числа запросов (HTML-запросы).

Нагрузочное тестирование веб-сервера с Nginx.

Для тестирования используется 2 виртуальные машины – одна с установленным и работающим Nginx, которой будут отсылаться запросы, другая будет отсылать эти самые запросы и делать выводы о производительности веб-сервера с Nginx. Примечание – Установка Nginx.

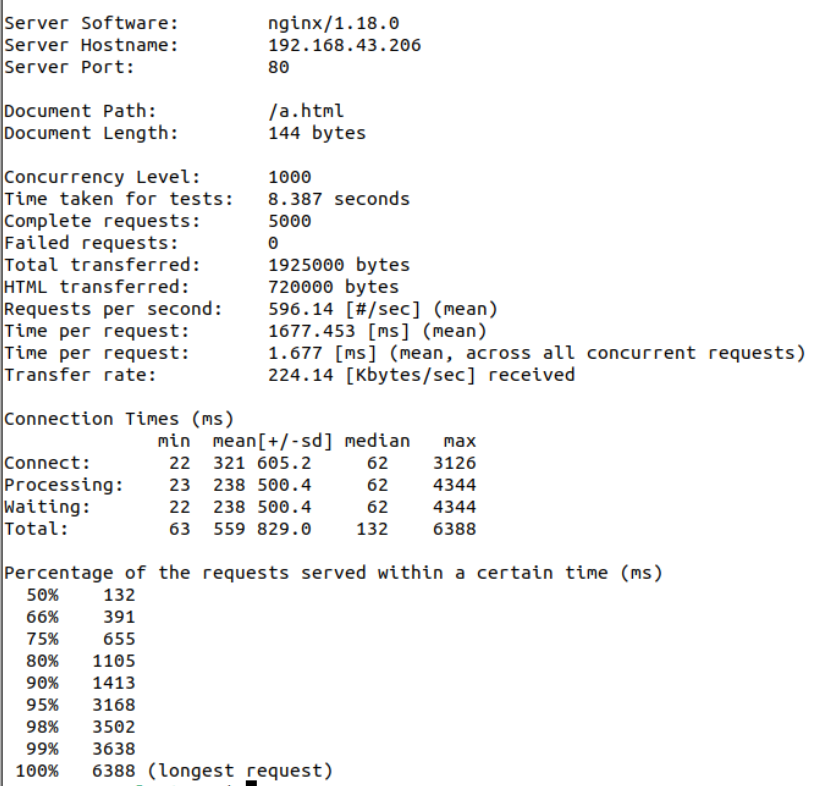
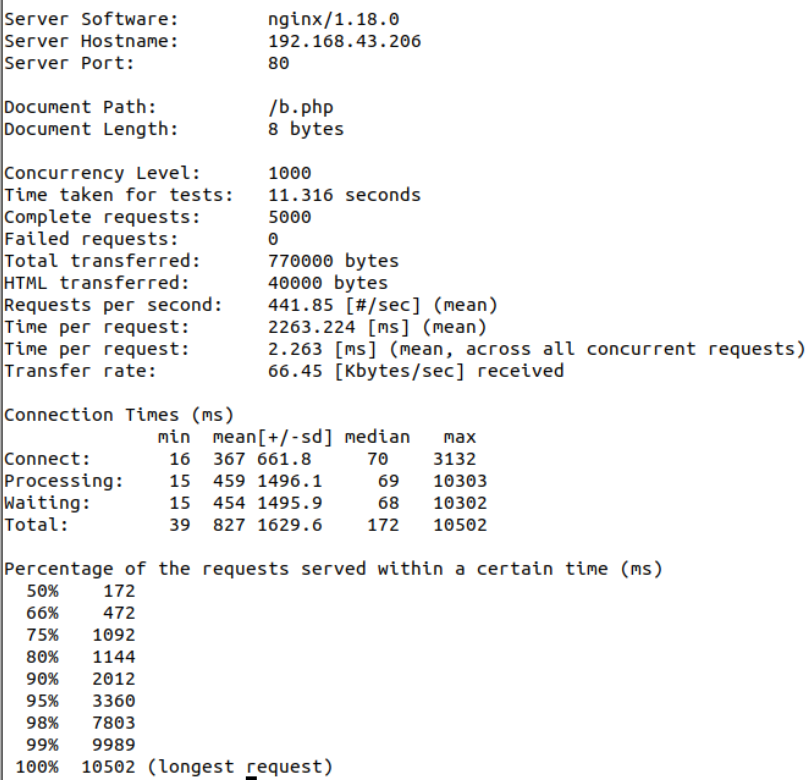


Рисунок 7 – Установка Nginx на две виртуальные машины.

Тестирование на PHP-запросы:

5. Определить максимальное число параллельных запросов, при котором сервер нас не будет блокировать.

6. Провести тест при использовании максимального числа запросов.

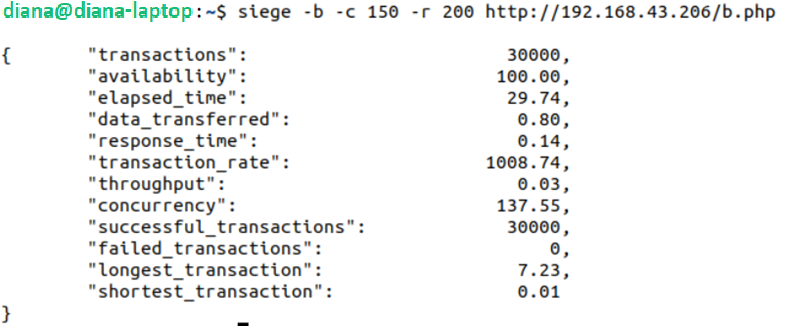


Рисунок 8 - Тест при использовании максимального числа запросов.

Тестирование на HTML-запросы:

7. Определить максимальное число параллельных запросов.

8. Провести тест при использовании максимального числа запросов.

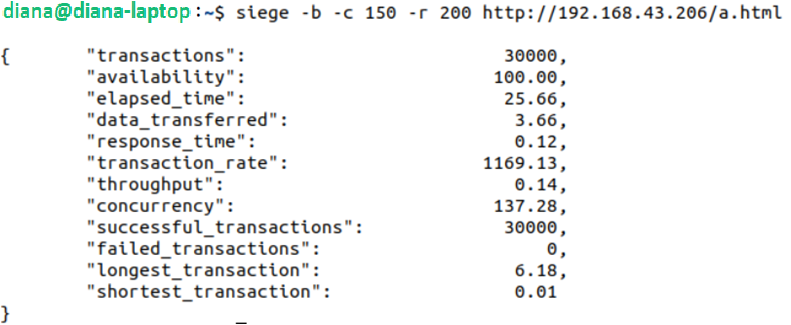


Рисунок 9 - Тест при использовании максимального числа запросов.

9. Провести сравнение результатов и сформировать выводы.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Максимальное число запросов | Запросы/сек | Время, затрачиваемое на запрос, мс | % успешных запросов |
| Apache | PHP | 200 | 29.88 | 0.14 | 100 |
| HTML | 200 | 25.58 | 0.13 | 100 |
| LB+  Apache | PHP | 1000 | 141.52 | 0.02 | 100 |
| HTML | 1000 | 24.91 | 0.04 | 100 |
| Nginx | PHP | 200 | 1008.74 | 0.14 | 100 |
| HTML | 200 | 1169.13 | 0.12 | 100 |
| LB+  Nginx | PHP | 1000 | 1246.05 | 0.04 | 100 |
| HTML | 1000 | 1314.28 | 0.05 | 100 |

**Вывод**

В ходе лабораторной работы были определена производительность web-серверов Apache и Nginx при помощи систем нагрузочного тестирования, а также был получен отказ в обслуживании.