Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра вычислительных технологий**

**ЛАБОРАТОРНАЯ РАБОТА №7**

**Дисциплина: Информационная безопасность**

Работу выполнил: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Романов В.В.

Направление подготовки: 02.03.02 Фундаментальная информатика и информационные технологии

Преподаватель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Шиян В.И.

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

**Ход работы:**

Для тестирования будем использовать 2 ВМ – одна с установленным и работающим Apache, вторая – отсылает запросы и делает выводы о производительности web-сервера.

1. Проведем тестирование на PHP-запросы.

Определим максимальное число параллельных запросов, при котором сервер не будет нас блокировать. На рисунках 1-3 показан процесс работы.

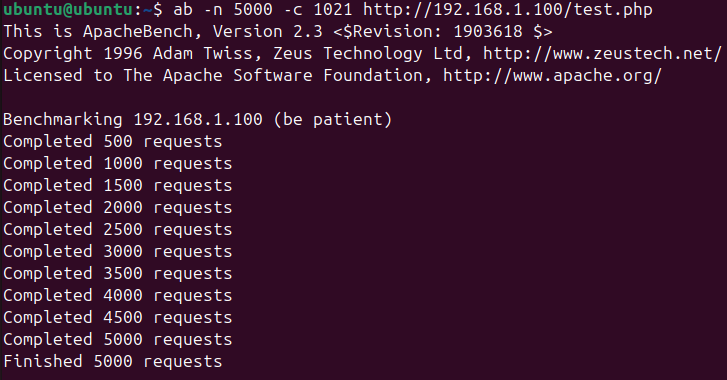


Рисунок 1 – Определение максимального числа запросов

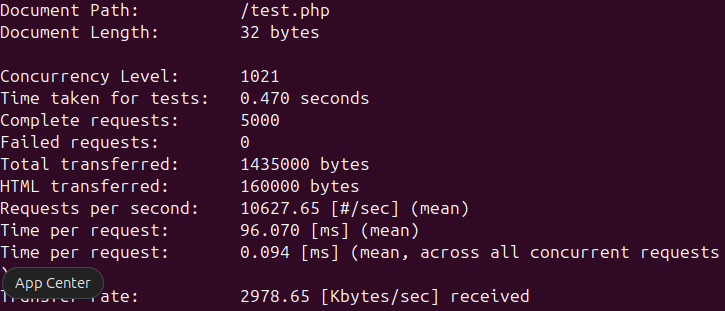


Рисунок 2 – Определение максимального числа запросов

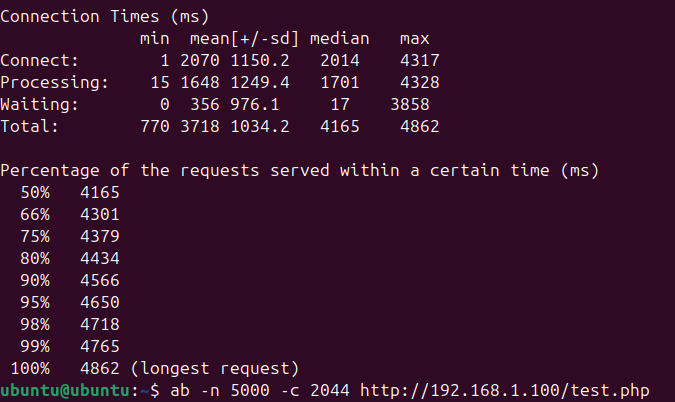


Рисунок 3 – Определение максимального числа запросов

1. Проведём тест при использовании максимального числа запросов. На рисунке 4 показан результат.

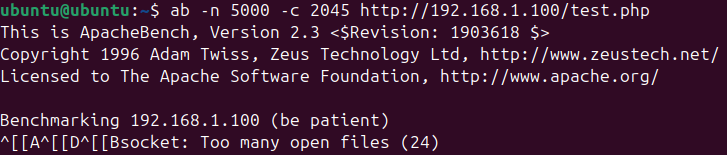


Рисунок 4 – Max число запросов, при котором происходит блокировка

1. Повторим теперь то же самое для HTML-запросов. Определим максимальное число параллельных запросов. Процесс представлен на рисунках 5-7.

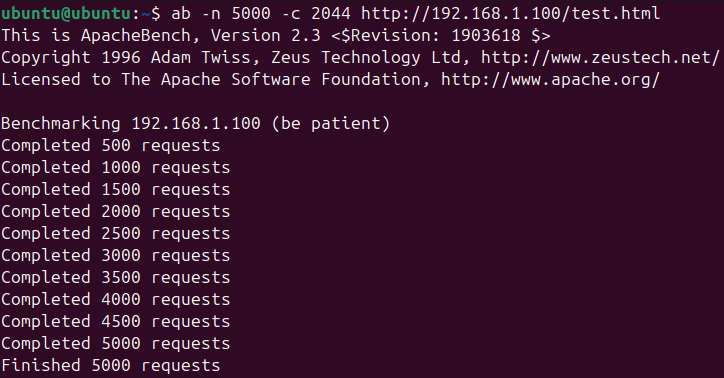


Рисунок 5 – Определение максимального числа запросов

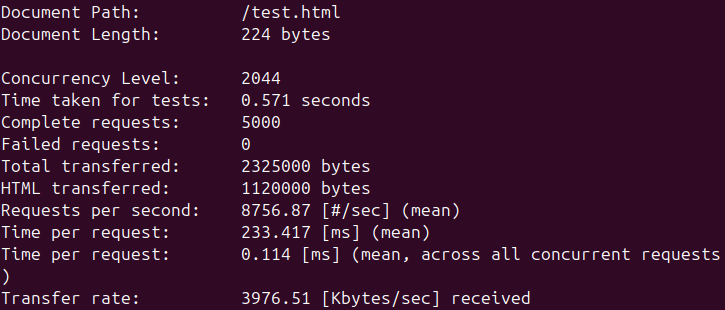


Рисунок 6 – Определение максимального числа запросов

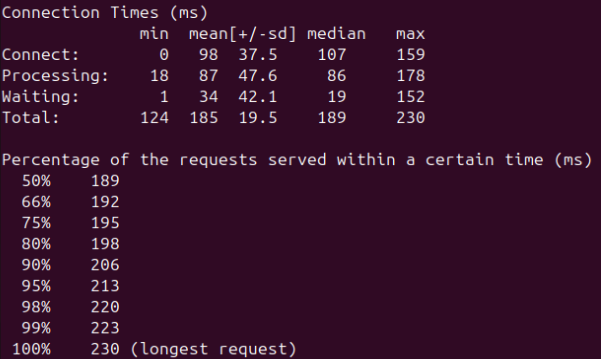


Рисунок 7 – Определение максимального числа запросов

1. Проведем тест при использовании максимального числа запросов. На рисунке 8 представлен результат тестирования.

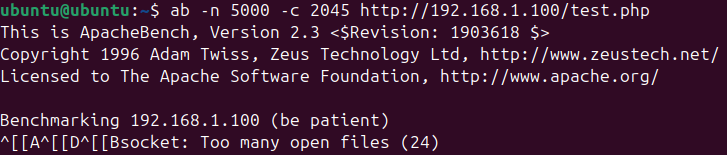


Рисунок 8 – Max число запросов, при котором происходит блокировка

1. Нагрузочное тестирование веб-сервера с Nginx.

Для тестирования используется 2 ВМ – одна с установленным и работающим Nginx, которой будут отсылаться запросы, другая будет отсылать эти запросы и делать выводы о производительности веб-сервера с Nginx. Определим максимальное число параллельных запросов, при котором сервер (не)будет нас блокировать. На рисунках 9-12 приведен результат.

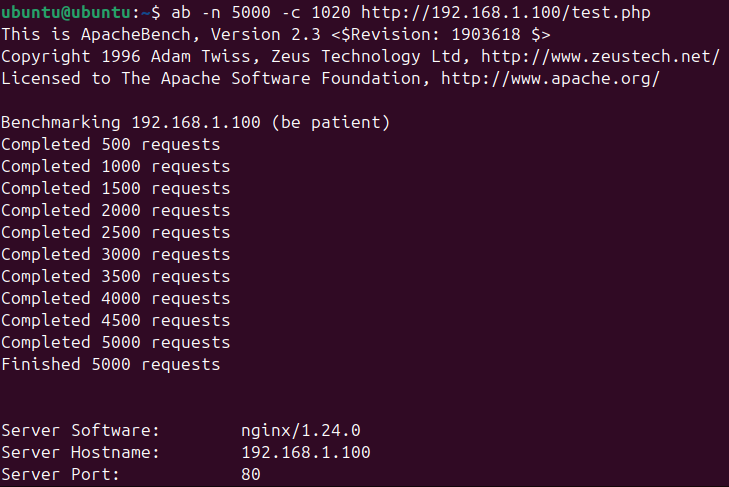


Рисунок 9 – Запросы

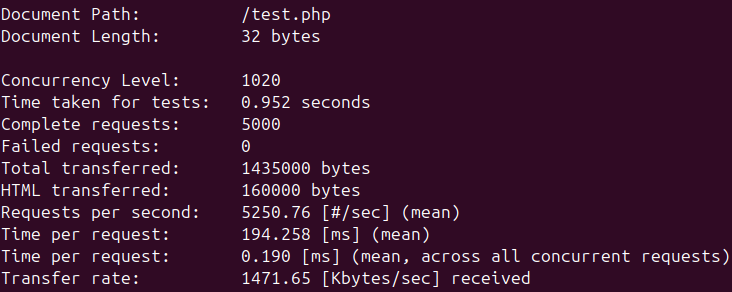


Рисунок 10 – Сведения о файле

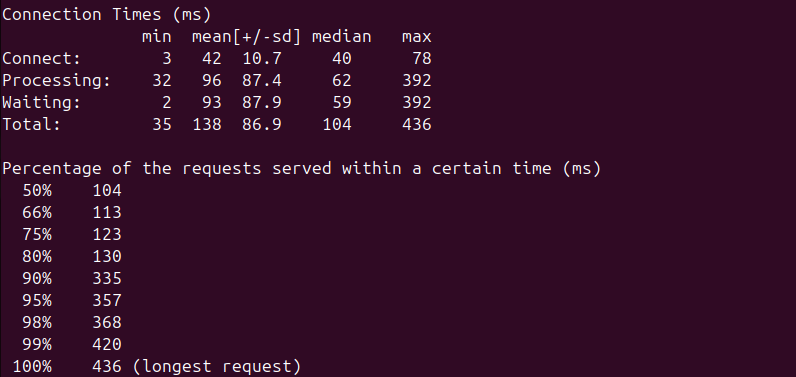


Рисунок 11 – Тестирование, при котором сервер не блокирует

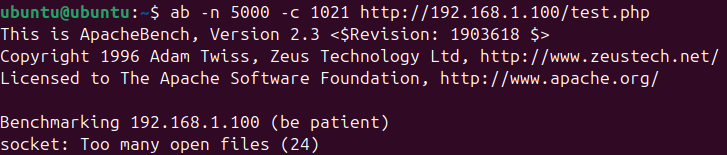


Рисунок 12 – Тестирование с максимальным числом запросов

1. Проведем тестирование на HTML-запросах. На рисунках 13-15 представлен результат.

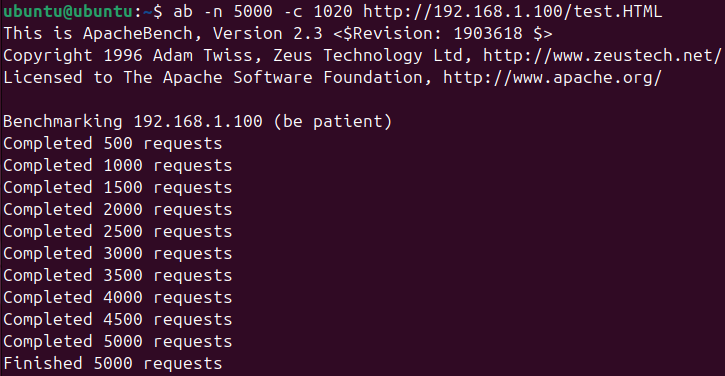


Рисунок 13 – Тестирование HTML-запросов

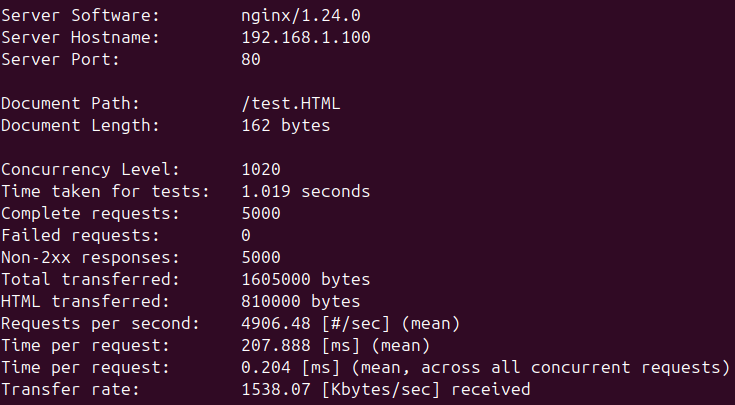


Рисунок 14 – Тестирование HTML-запросов

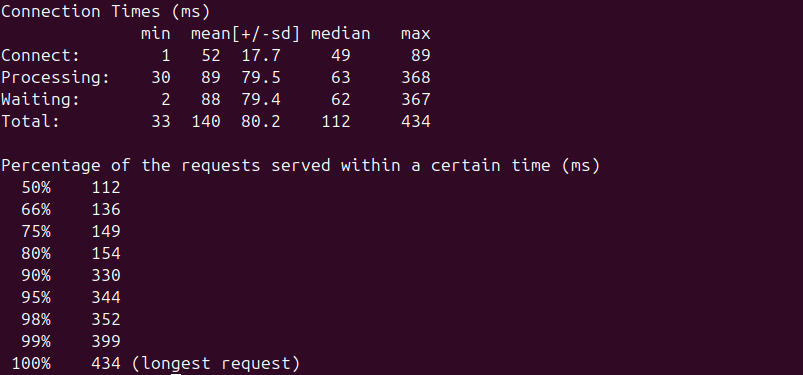


Рисунок 15 – Максимальное число параллельных запросов

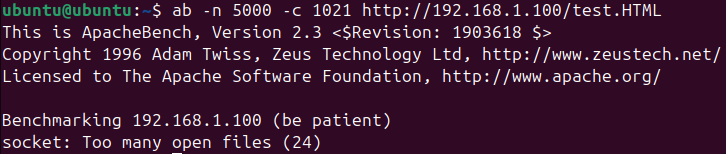


Рисунок 16 – Тест при использовании максимального числа запросов

1. Нагрузочное тестирование веб-серверов Apache с балансировщиком нагрузки. Проведем тестирование на PHP-запросах. На рисунках 17-22 приведены результаты работы.

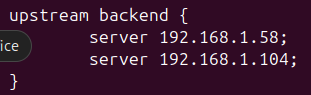


Рисунок 17 – Настройка сервера балансировщика на основе Nginx



Рисунок 18 – PHP-запросы

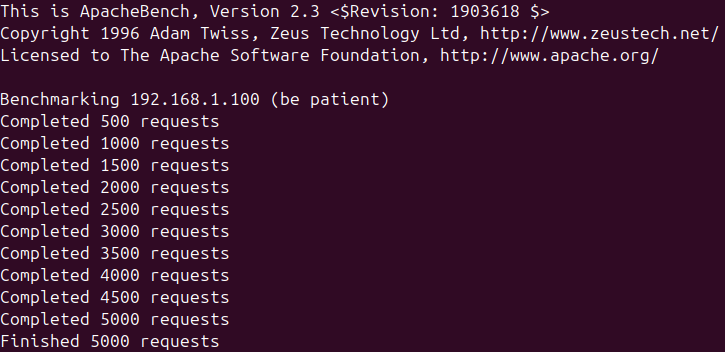


Рисунок 19 – Запросы

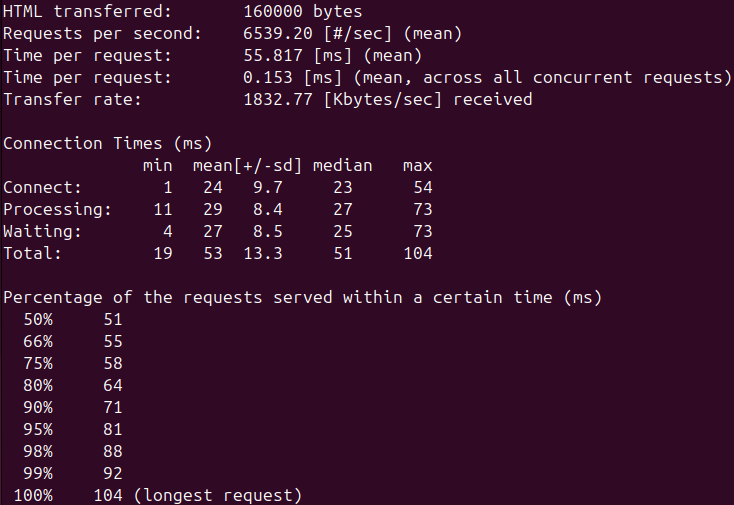


Рисунок 20 – Результат тестирования PHP-запросы



Рисунок 21 – HTML-запросы

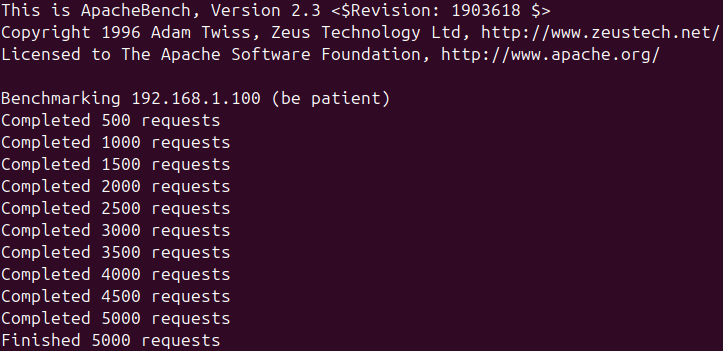


Рисунок 22 – Запросы

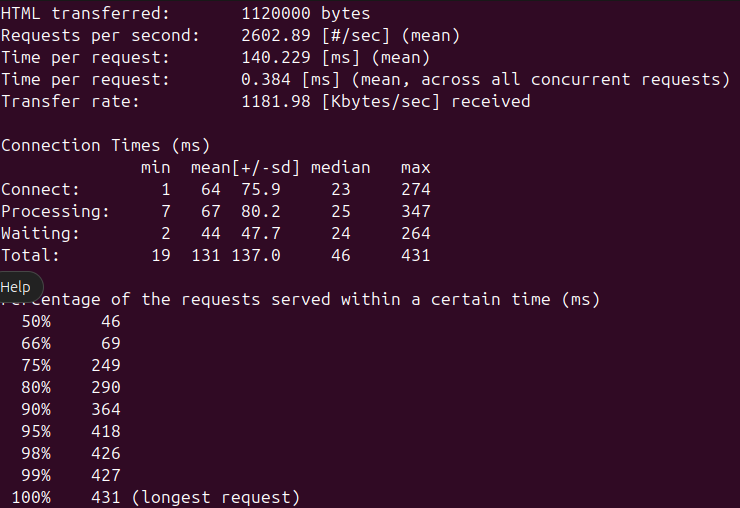


Рисунок 23 – Результат тестирования HTML-запросами

1. Проведем тестирование с максимальным числом PHP-запросов на случай, если оба сервера на другом конце – Nginx. На рисунках 24-26 изображено вышесказанное.



Рисунок 24 – Тестирование с Max числом PHP-запросов

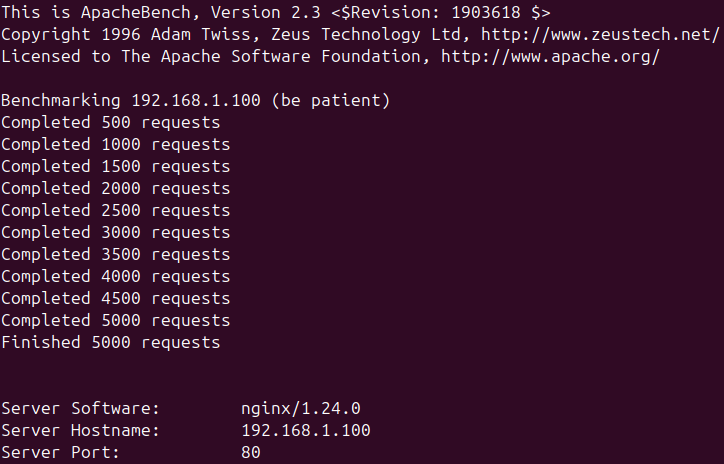


Рисунок 25 – Запросы

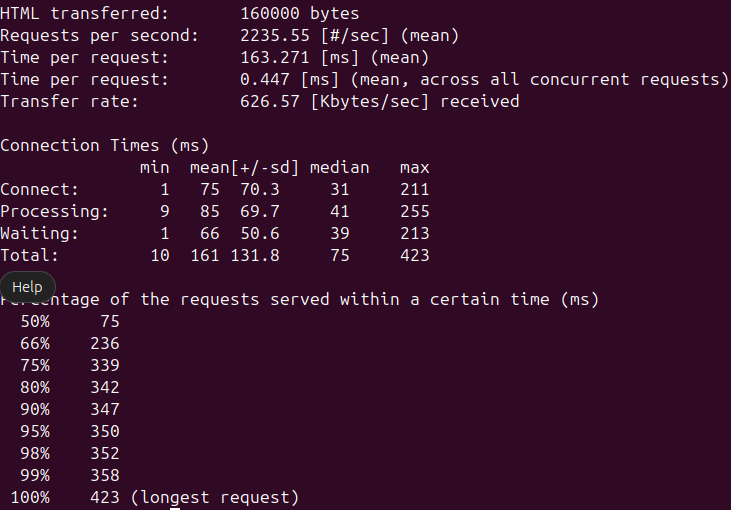


Рисунок 26 – Результат тестирования PHP-запросами

1. Проведем тестирование с максимальным числом HTML-запросов на случай, если оба сервера на другом конце – Nginx. На рисунках 27-29 представлен результат.



Рисунок 27 – HTML-запросы

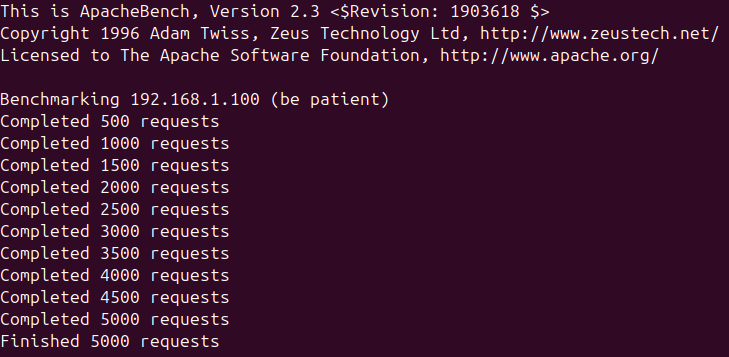


Рисунок 28 – HTML-запросы

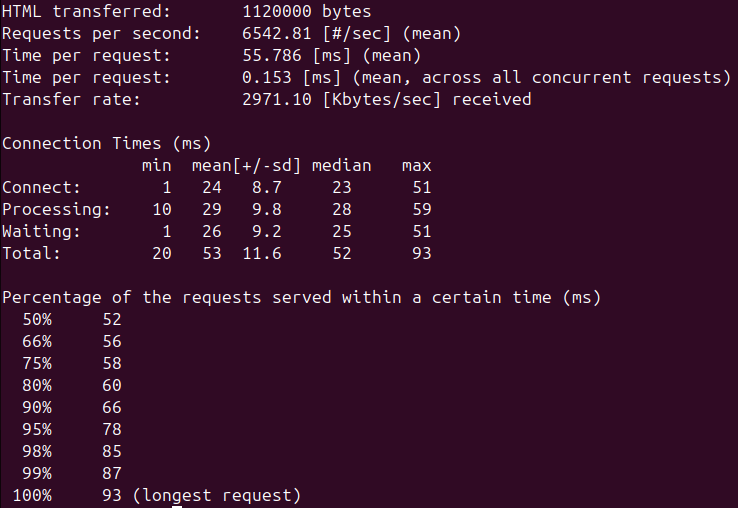


Рисунок 29 – HTML-запросы

1. Итоговая таблица с результатами.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Максимальное число запросов | Запросы/сек | Время, затрачиваемое на запрос, мс | % успешных запросов |
| Apache2 | PHP | 2045 | 10627 | 96 | 100% |
| HTML | 2045 | 8756 | 233 | 100% |
| LB + Apache2 | PHP | 365 | 2602 | 140 | 100% |
| HTML | 365 | 2235 | 163 | 100% |
| Nginx | PHP | 1021 | 5250 | 194 | 100% |
| HTML | 1021 | 4906 | 207 | 100% |
| LB + Nginx | PHP | 365 | 4344 | 55 | 100% |
| HTML | 365 | 1785 | 204 | 100% |

Анализируя полученные результаты в таблице, можно сделать вывод, что Nginx с балансировщиком нагрузки может быть оптимальным выбором для высоконагруженных PHP-запросов, ведь он обеспечивает высокую пропускную способность и низкое время отклика.

**Вывод:** с помощью систем нагрузочного тестирования определил производительность web-серверов Apache и Nginx, а также добился отказа в обслуживании.