**Реализация алгоритмов балансировки**

Результаты анализа алгоритмов балансировки:

1. Round Robin распределяет запросы равномерно по серверам с учетом их весов. Например, если один из серверов имеет *weigth=5*, то 5 из 6 запросов будут переданы ему, а один другому серверу.
2. Least Connections отправляет запрос на сервер с наименьшим количеством активных подключений с учетом их весов.
3. IP-hash отправляет запрос на сервер, который определяется по IP-адресу клиента.
4. Hash отправляет запрос на сервер, который определяется по заданному пользователем ключу.
5. Least Time отправляет запрос на сервер с минимальной средней задержкой и минимальным количеством активных подключений.
6. Random отправляет запрос на случайно выбранный сервер.

**Реализация отказоустойчивости**

Для проверки функционала реализации отказоустойчивости при отключении одного из серверов зададим следующие параметры в файле nginx\_round\_robin:

*max\_fails=3 fail\_timeout=30s*

Это значит, что если NGINX не смог отправить запрос 3 раза в течение 30 секунд, то NGINX помечает сервер, как недоступный. Результатом на практике можно заметить. что в случае недоступности одного из серверов. NGINX направляет запрос на другой сервер.

**Нагрузочное тестирование**

Для анализа результатов нагрузочного тестирования с использованием Apache benchmark выполним следующие шаги:

1. Извлечем образ:

*docker pull jordi/ab*

1. Запустим стек приложений:

*docker-compose up -d --build*

1. Далее выполним нагрузочное тестирование, в котором осуществляется 1000 запросов к серверу, одновременно 100 запросов максимум:

*docker run --rm jordi/ab -k -c 100 -n 1000 http://192.168.99.100:8080/*

1. Анализ результатов

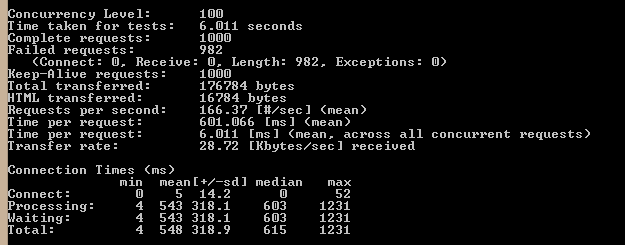


Рис. 1. Результаты анализа

По рис. 1 можно сделать следующие основные выводы:

* 1. выполнено 1000 запросов;
  2. сервер возвращает запросы со скоростью 166.37 запросов/сек.
  3. самый быстрый запрос составил 4 мс, самый медленный 1231 мс.