Практическая работа № 5.
Нагрузочное тестирование с помощью утилиты AB (Apache Benchmark).
Конфигурирование веб-сервера nginx.

AB - это утилита для тестирования производительности вашего веб-сервера. Она создана для того, чтобы вы могли определить производительность вашей текущей настройки веб-сервера. В первую очередь, AB показывает сколько запросов в секунду способен обслужить ваш веб-сервер

Установка Apache Benchmark

AB входит в пакет apache2-utils. Для установки необходимо выполнить команду:

```
apt-get install apache2-utils
```

Синтаксис команд имеет следующий вид:

```
ab [options] [http[s]://]hostname[:port]/path
```

Options:

`-A auth-username:password` - передать данные для базовой аутентификации. То есть, фактически можно тестировать даже если доступ закрыт с помощью базовой аутентификации (htpasswd).

`-c concurrency` - количество параллельных запросов в единицу времени. По умолчанию, один реквест в единицу времени (можно считать, что в секунду).

`-C cookie-name=value` - добавлять куки. Задается в виде пары имя=значение. Это поле можно повторять.

`-f protocol` - задает SSL/TLS протокол (SSL2, SSL3, TLS1, or ALL).

`-h` - отобразить краткую справку по параметрам

`-k` - включить KeepAlive, то есть осуществлять множество запросов в течение одной HTTP-сессии. По умолчанию данная возможность отключена.

`-n requests` - количество запросов, которое необходимо выполнить в течение сессии тестирования. По умолчанию, выполняется только один запрос, что не дает общей картины.

`-q` - подавляет вывод некоторых сообщений о процессе тестирования.

`-t timelimit` - максимальное количество секунд, которое необходимо затратить на тестирование. Это подразумевает значание параметра -n равное 50000. По умолчанию временной лимит не установлен.

`-v verbosity` - устанавливает уровень "разговорчивости": 4 и выше отображает информацию о заголовках, 3 и выше - информацию о кодах ответа (404, 200 и т. д.), 2 и выше - выводить предупреждения и прочую информацию.

Последовательные запросы

Для того, чтобы организовать очередь из 1000 запросов (1 запрос в одну единицу времени), необходимо выполнить команду:

```
ab -n 1000 <URL>
```

При успешном выполнение вывод команды выглядит так:

```
Benchmarking mysite.loc (be patient)
Completed 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Completed 500 requests
Completed 600 requests
Completed 700 requests
Completed 800 requests
Completed 900 requests
Completed 1000 requests
Finished 1000 requests
```

Далее будет идти информация о веб-сервере, а также размере и типе загружаемого документа:

```
Server Software:        nginx/1.13.7
Server Hostname:        mysite.loc
Server Port:            80

Document Path:          /index.html
Document Length:        612 bytes
```

Результаты тестирования:

Concurrency Level:	1	Количество одновременно-отправляемых запросов
Time taken for tests:	0.659 seconds	Время тестирования
Complete requests:	1000	Успешные запросов
Failed requests:	0	Безуспешные запросы
Total transferred:	845000 bytes	Объем переданных данных
HTML transferred:	612000 bytes	HTML-контент
Requests per second:	1517.81 [#/sec] (mean)	Запросов в секунду
Time per request:	0.659 [ms] (mean)	Время на один запрос
Transfer rate:	1252.49 [Kbytes/sec] received	Скорость обмена данными

Параллельные(одновременные) запросы

Для того, чтобы организовать очередь из 1000 запросов(1000 параллельных запросов в одну единицу времени) необходимо выполнить команду:
ab -n 1000 -c 1000 <URL>. Вывод команды имеет аналогичную структуру, как и в случае последовательных запросов.

Конфигурирование веб-сервера

Производительность веб-сервера в первую очередь зависит от железа на котором он работает, пропускной способности сети и настроек. Файл глобальных настроек - /etc/nginx/nginx.conf.

Директивы

worker_processes - задает число рабочих процессов. Оптимальное значение зависит от множества факторов, включая (но не ограничиваясь ими) число процессорных ядер, число жёстких дисков с данными и картину нагрузок. Если затрудняетесь в выборе правильного значения, можно начать с установки его равным числу процессорных ядер (значение “auto” пытается определить его автоматически).

worker_connections - задаёт максимальное число соединений, которые одновременно может открыть рабочий процесс. Следует иметь в виду, что в это число входят все соединения (в том числе, например, соединения с проксируемыми серверами), а не только соединения с клиентами. Стоит также учитывать, что фактическое число одновременных соединений не может превышать действующего ограничения на максимальное число открытых файлов, которое можно изменить с помощью worker_rlimit_nofile.

Если worker_processes и worker_connection настроены должным образом, а нагрузочное тестирование падает на маленьком кол-ве запросов - смотрим логи, скорее всего проблема с worker_rlimit_nofile (ошибка too many files open).

Максимальное количество соединений, которые Nginx может обслуживать одновременно определяются произведением двух параметров:
Всего соединений = worker_processes * worker_connection. На рис. 1 схематично показано, что есть worker(worker processes)/worker connection.

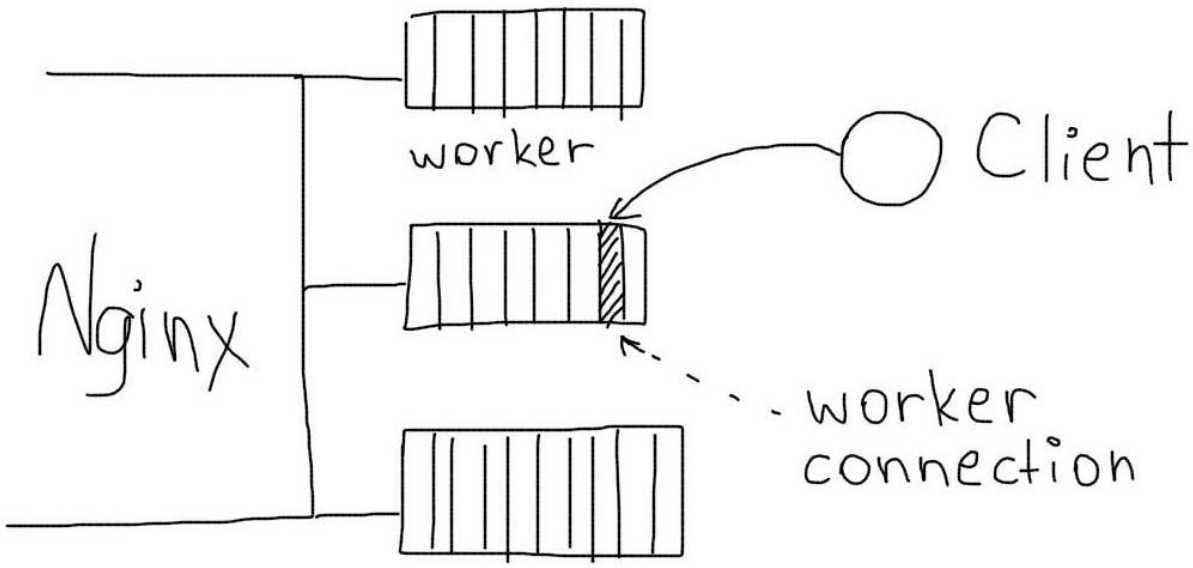

Рисунок 1. Worker_process & worker_connection

Задание на лабораторную работу.

1. Найти максимально возможное кол-во соединений на стандартной конфигурации nginx, с помощью утилиты для нагрузочного тестирование ab, необходимо тестировать с флагами “-n <any_number> -c <any_number>”.

2. Заставить сервер обрабатывать 30000 запросов, при частоте 1000 запросов/единица времени(параметр -c).

Контрольные вопросы.

1. Какие ещё виды тестирования существуют помимо нагрузочного?
2. В чем различие файла настроек /etc/nginx/conf.d/<name_vhost>.conf от /etc/nginx/nginx.conf ?
3. В чем различие директив worker_processes и worker_connection?

Полезные ресурсы.

1. Все про nginx на Хайлоаде. https://ruhighload.com/nginx