

При разработке веб приложений, будь то личные блоги, интернет магазины или многофункциональные порталы, полезно знать, какую нагрузку они смогут выдерживать. Основной задачей любого тестирования производительности сайта является понимание его устойчивости к нагрузкам, которые могут появляться не только из-за большого количества посетителей онлайн, но и являться следствием некорректной настройки сервера, неправильной работы скриптов или действиями злоумышленников (DOS, DDOS).

Ab — небольшая утилита, входящая в пакет apache2-utils, но предустановленная во многих современных дистрибутивах. Она не требует привилегированного доступа к системе и на фоне более гибких конкурентов крайне неприхотлива.

Если в вашей системе ab еще не установлен:

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

Синтаксис команды следующий:

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

Самыми важными ключами для любого тестирования являются ключ n — количество запросов страницы и ключ c — количество конкурентных запросов. Запустим утилиту с этими ключами:

```
ab -c 10 -n 100 https://example.org/
```

После завершения работы утилиты мы увидим подобный вывод:

```
This is ApacheBench, Version 2.3 <$Revision: 655654 $>  
Copyright 1996 Adam Twiss, Zeus Technology Ltd,  
http://www.zeustech.net/  
Licensed to The Apache Software Foundation,  
http://www.apache.org/
```

```
Benchmarking google.com (be patient).....done
```

```
Server Software: nginx/1.4.2  
Server Hostname: example.org  
Server Port: 80
```

```
Document Path: /  
Document Length: 57606 bytes
```

```
Concurrency Level: 10  
Time taken for tests: 0.190 seconds  
Complete requests: 100
```

Failed requests: 0
Write errors: 0
Keep-Alive requests: 100
Total transferred: 5784600 bytes
HTML transferred: 5760600 bytes
Requests per second: 526.32 [#/sec] (mean)
Time per request: 19.000 [ms] (mean)
Time per request: 1.900 [ms] (mean, across all concurrent requests)
Transfer rate: 29732.17 [Kbytes/sec] received

Connection Times (ms)
min mean[+/-sd] median max
Connect: 0 0 1.1 0 4
Processing: 12 18 2.7 19 23
Waiting: 3 6 1.3 6 8
Total: 12 18 3.3 19 27

Percentage of the requests served within a certain time (ms)
50% 19
66% 20
75% 20
80% 20
90% 24
95% 25
98% 26
99% 27
100% 27 (longest request)

Рассмотрим вывод построчно:

Server Software — информация о frontend сервере, переданная в http head.

Server Hostname — имя тестируемого хоста.

Server Port — порт подключения.Document Path — относительный путь без названия хоста.

Document Length — длина возвращенного документа.Concurrency Level — количество конкурентных запросов из ключа c.

Time taken for tests — время, затраченное на тест.

Complete requests — количество выполненных запросов. Если тест прошел без обрывов, значение совпадает с ключом n.

Failed requests — количество запросов с ошибками. В рамках теста ошибки разделяются на четыре типа — ошибки соединения, ошибки передачи, ошибки длины возвращенных данных, исключения). Про ошибки длины стоит добавить, что если тестируемая страница меняет свое содержимое при обновлении, т.е. является динамической, добавляйте ключ -l.

Write errors — ошибки записи. Сюда также часто сыпятся ошибки Broken pipe.

Non-2xx responses — ответы с не 2xx кодами. Т.е. ошибочные либо умышленно, либо в результате проблем на сервере.

Total transferred — общий объем переданных данных

HTML transferred — объем переданных HTML данных

Requests per second— среднее количество обработанных в секунду запросов.

Time per request — среднее время обработки запроса.

Time per request — среднее время обработки запроса учитывая конекурентность.

Transfer rate — скорость передачи данных. Connection Times (ms) — время соединения в миллисекундах

Столбцы: min — минимальное время, mean[+/-sd] — среднеквадратическое отклонение, median — среднее время, max — максимальное время.

Строки: connect — время соединения с сервером, processing — время обработки запроса, waiting — полное время ожидания, включая processing и время ожидания в очереди, total — общее время выполнения запроса.

Percentage of the requests served within a certain time (ms) — доля запросов, выполненных в определенное время. В нашем случае 50% всех запросов было обработано за 19 миллисекунд, а интервал до 27 миллисекунд выолнились все 100% запросов.

50% 19

66% 20

75% 20

80% 20

90% 24

95% 25

98% 26

99% 27

100% 27 (longest request)

Основными показателями эффективности работы сайта можно выделить Requests per second, Time per request и не сильный разрыв в графике долгих запросов. Важно помнить, что географическая удаленность до сервера играет важную роль при анализе показаний, так что если ваш сервер располагается в дальних и не очень странах, не стоит расстраиваться из-за времени выполнения до 200 мс, главное — отсутствие странных перекосов последней секции.

Ниже представлен список поддерживаемых опций ab. Их использование во многом способствует конкретизации тестов, так что не стоит ими пренебрегать

- n requests количество запросов страницы.
- c concurrency количество конкурентных запросов.
- t timelimit максимальное время теста
- s timeout максимальное время на один запрос. По умолчанию 30 секунд.
- b window size размер TCP буфера в байтах
- V address адрес для исходящих подключений
- p postfile Файл, содержащий данные для POST. Не забудьте также установить -T
- u putfile Файл, содержащий данные для PUT. Не забудьте также установить -T
- T content-type HTTP заголовок для методов POST/PUT. По умолчанию text/plain
- w Вывести результат в виде HTML.
- C attribute Добавить cookie, например 'Apache=1234'
- H attribute добавить произвольную строку заголовка, например

'Accept-Encoding: gzip'

-A attribute Использовать Basic WWW Authentication, например -A user pass
-P attribute Использовать аутентификацию на Proxu, например -P прохуuser прохурpass
-X проху:port Указать Proxu сервер
-V Отобразить версию ab
-k Использовать KeepAlive
-l Разрешить изменяемую длину документа. Используйте для динамических страниц.
-g filename Сохранить результат в формате gnuplot.
-e filename Сохранить результат в CSV.
-r Не прекращать тест при ошибках передачи.
-h Отобразить справку
-f protocol Указать протокол. (SSL3, TLS1, TLS1.1, TLS1.2 или ALL)

Стандартный сценарий тестирования — последовательный запуск теста ab с увеличением конкурентных запросов и мониторинг зависимости полученных значений. Если вы хотите просто протестировать работу вашего PHP, достаточно создать файл с содержанием phpinfo(). Если при наращивании конкурентных запросов вы получаете ошибку:

```
socket: Too many open files (24)
```

достаточно выполнить:

```
ulimit -n 8192
```

и лимит подключений увеличится до 8192.

—
Источник — admins.su