You can benchmark Apache, IIS and other web server with apache benchmarking tool called ab. Recently I was asked to performance benchmarks for different web servers.

It is true that benchmarking a web server is not an easy task. From how to benchmark a web server:

First, benchmarking a web server is not an easy thing. To benchmark a web server the time it will take to give a page is not important: you don't care if a user can have his page in 0.1 ms or in 0.05 ms as nobody can have such delays on the Internet.

What is important is the average time it will take when you have a maximum number of users on your site simultaneously. Another important thing is how much more time it will take when there are 2 times more users: a server that take 2 times more for 2 times more users is better than another that take 4 times more for the same amount of users."

Here are few tips to carry out procedure along with an example:

Apache Benchmark Procedures

- You need to use same hardware configuration and kernel (OS) for all tests
- You need to use same network configuration. For example, use 100Mbps port for all tests

- First record server load using top or uptime command
- Take at least 3-5 readings and use the best result
- After each test reboot the server and carry out test on next configuration (web server)
- Again record server load using top or uptime command
- Carry on test using static html/php files and dynamic pages
- It also important to carry out test using the Non-KeepAlive and KeepAlive (the Keep-Alive extension to provide long-lived HTTP sessions, which allow multiple requests to be sent over the same

TCP connection) features

- Also don't forget to carry out test using fast-cgi and/or perl tests

Webserver Benchmark Examples:

Let us see how to benchmark a Apache 2.2 and lighttpd 1.4.xx web server.

Static Non-KeepAlive test for Apache web server

i) Note down server load using uptime command
$ uptime

ii) Create a static (small) html page as follows (snkpage.html) (assuming that server IP is 202.54.200.1) in /var/www/html (or use your own webroot):

Webserver test This is a webserver test page.

Login to Linux/bsd desktop computer and type following command:
$ ab -n 1000 -c 5 http://202.54.200.1/snkpage.html
Where,

- -n 1000: ab will send 1000 number of requests to server 202.54.200.1 in order to perform for the benchmarking session
- -c 5 : 5 is concurrency number i.e. ab will send 5 number of multiple requests to perform at a time to server 202.54.200.1

For example if you want to send 10 request, type following command:
$ ab -n 10 -c 2 http://www.somewhere.com/

Output:

This is ApacheBench, Version 2.0.41-dev apache-2.0 Copyright (c) 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/ Copyright (c) 1998-2002 The Apache Software Foundation, http://www.apache.org/ Benchmarking www.cyberciti.biz (be patient).....done Server Software: Server Hostname: www.somewhere.com Server Port: 80 Document Path: / Document Length: 16289 bytes Concurrency Level: 1 Time

taken for tests: 16.885975 seconds Complete requests: 10 Failed requests: 0 Write errors: 0 Total transferred: 166570 bytes HTML transferred: 162890 bytes Requests per second: 0.59 [#/sec] (mean) Time per request: 1688.597 [ms] (mean) Time per request: 1688.597 [ms] (mean, across all concurrent requests) Transfer rate: 9.59 [Kbytes/sec] received Connection Times (ms) min mean[+/-sd] median max Connect: 353 375 16.1 386 391 Processing: 1240 1312 52.1 1339 1369 Waiting: 449 472 16.2 476 499 Total: 1593 1687 67.7 1730 1756 Percentage of the requests served within a certain time (ms) 50% 1730 66% 1733 75% 1741 80% 1753 90% 1756 95% 1756 98% 1756 99% 1756 100% 1756 (longest request)

Repeat above command 3-5 times and save the best reading.

Static Non-KeepAlive test for lighttpd web server

First, reboot the server:
reboot

Stop Apache web server. Now configure lighttpd and copy /var/www/html/snkpage.html to lighttpd webroot and run the command (from other linux/bsd system):
$ ab -n 1000 -c 5 http://202.54.200.1/snkpage.html

c) Plot graph using Spreadsheet or gnuplot.

How do I carry out Web server Static KeepAlive test?

Use -k option that enables the HTTP KeepAlive feature using ab test tool. For example:
$ ab -k -n 1000 -c 5 http://202.54.200.1/snkpage.html

Use the above procedure to create php, fast-cgi and dynmic pages to benchmarking the web server.

Please note that 1000 request is a small number you need to send bigger (i.e. the hits you want to test) requests, for example following command will send 50000 requests :
$ ab -k -n 50000 -c 2 http://202.54.200.1/snkpage.html

How do I save result as a Comma separated value?

Use -e option that allows to write a comma separated value (CSV) file which contains for each percentage (from 1% to 100%) the time (in milliseconds) it took to serve that percentage of the requests:

```
$ ab -k -n 50000 -c 2 -e apache2r1.cvs http://202.54.200.1/snkpage.html
```

How do I import result into excel or gnuplot programs so that I can create graphs?

Use above command or -g option as follows:

```
$ ab -k -n 50000 -c 2 -g apache2r3.txt http://202.54.200.1/snkpage.html
```

Put following files in your webroot (/var/www/html or /var/www/cgi-bin) directory. Use ab command.

Sample test.php file

Php test with phpinfo()

Run ab command as follows:

```
$ ab -n 500 -c 5 http://202.54.200.1/test.php
```

Sample test.pl (perl) file

```
#!/usr/bin/perl $command=`perl -v`; $title = "Perl Version"; print "Content-type: text/htmlnn
"
;
print
"$title
n
n
n
"
;
print
"
```

$title

```
n"; print $command; print "nn";
```

Run ab command as follows:

```
$ ab -n 3000 -c 5 http://202.54.200.1/cgi-bin/test.pl
```

Sample psql.php (php+mysql) file

Php+MySQL

Run ab command as follows:

```
$ ab -n  1000 -c 5 http://202.54.200.1/psql.php
```