

HP / March 03, 2009 11:13PM

[Apache 內建的Web壓力測試工具 ApacheBench \(ab, ab2\)](#)官方使用說明：[ab - Apache HTTP server benchmarking tool](#) (Apache 2.2)

簡易說明：

ApacheBench 的使用很簡單，基本上有裝 Apache 就會有，直接以 ab 或 ab2 作為指令使用就可以了。(注意有些 Linux 版本的指令是 ab2)

情況一：公司要測試網站能否達到「首頁同時1,000人連線，平均反應時間 5 秒以下。」

此時使用以下指令即可：

```
Shell> ab -k -c 1000 -n 1000 http://yoursite.com/index.html
```

參數：

-k 是 HTTP 2.0 的 Keep Alive，在短時間的測試下 connection 有無 Keep Alive 差別不是很大。

-c 是指 concurrent，顧名思義其後的數字是「同時連線」的 connection 數。

-n 是指整個 test 的 request number，此處設為和 -c 一樣是指同時1000個人連線測完一次就中止。

底下是範例結果，「粗黑體」的數字 (1687 ms) 就是「平均反應時間」。這邊是 1.687 秒的意思。基本上如果是「純 HTML」的頁面，伺服器又不會太爛，concurrency 1000 的反應時間都不會太久(也許就像此處的 2 秒以下)。

當然，要注意「同時1000個連線」這件事理論上是很難模擬的，此處只是「盡力」。1~2 秒間能後送出 1,000 個 requests 就很厲害了。

```
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
```

--

底下節錄的文章來源：[Stress testing a site with Apache's ab tool. part 1](#) (很不錯的介紹)

以下節錄有關「送出表格」方面的壓力測試：

What if the site is behind a login that is an http form and not an htaccess form? That is where POST variables come into play, it is a little trickier but doable. Here is the command I use to hit a page behind a login.

```
ab -p ~/Documents/post.txt -T application/x-www-form-urlencoded -e http://example.com/manager/
```

Here is what my post.txt file contains:

```
email=test@domain.com&password=testing&submit=Login
```

Next we need to get these results to a file so we can work with them and have a record of progress as we tweak our site. It can be done with the "-e" parameter. An example is below:

```
ab http://example.com/search.php?q=Trees -e ~/Documents/results1.csv
```

What that does is hits the search page and puts the results in the results1.csv file. It is useful to put the file at the end of the command since you will want to change the filename on each run to keep a history of your results.

The csv file is separated into two columns. The first is the percentage complete and the second is the time in ms that it took to get to that percentage.

The first thing you want to look at is the zero percent. This is your latency, it is the amount of time it takes to fulfill the first request. You want this number to be as low as possible. I am running my site on a development server on my local network so the times will be lower but at the end I will show the results from the testing server.

```
ab -p ~/Documents/aci_post.txt -T application/x-www-form-urlencoded -e ~/Documents/aci_output.csv -kc 10 -t 10 http://aci/manager/makes.php
```

Edited 5 time(s). Last edit at 04/30/2010 11:48PM by HP.
