



網頁伺服器服務可用性偵測系統

報告者：李政賢



2009. 04. 10

大綱

- 前言
- 相關的網路品質狀態量測方法
 - SNMP(Simple Network Management Protocol)
 - Ping
 - TCP Socket
 - SWPM(Simply Web Traffic Meter)
- 系統架構與實作
 - cURL
 - PHP/cURL自動化登入量測程式設計
 - 以moodle入口網站為例
- 量測結果與結論
- 未來工作

前言

- 網頁伺服器服務
 - 伺服器本身或網路連線出了問題，使用者將無法正常使用服務
 - 網頁伺服器服務在應用層發生問題，傳統偵測方式無法偵出
 - N-tier架構之網頁伺服器服務發生問題不易偵測
- 本文提出一套模擬client實際使用網頁伺服器服務之偵測系統可用性方法
 - 可以更精確貼近使用者使用系統感受
 - 使用應用層（Application Layer）的偵測技術，能夠偵測n-tier架構的網路服務是否運作正常

相關的網路品質狀態量測方法

- SNMP(Simple Network Management Protocol)
 - 優點
 - 運作於網路層(Network Layer)利用內建SNMP模組取得網路設備相關資訊數據
 - 管理功能強大、設備支援度高
(大多數中高階網路設備都有支援)
 - 可針對想要監控的伺服器安裝SNMP agent
 - 缺點
 - 若管理的伺服器數量龐大，安裝SNMP agent的工作將變得耗費時間
 - 啟用SNMP可能有潛在的安全性風險
 - 當應用層網路服務異常時無法偵測反應

相關的網路品質狀態量測方法

- Ping

- 優點

- 簡單易用的一種瞭解網路連通狀況方式
 - 利用發送ICMP Echo Request封包方式，得知封包來回時間

- 缺點

- 基於安全性考量，越來越多網路管理者關閉設備對ICMP封包的回應
 - 在網路層中運作，若異常狀況發生在更上層亦無法偵測出來

相關的網路品質狀態量測方法

- TCP socket 連線

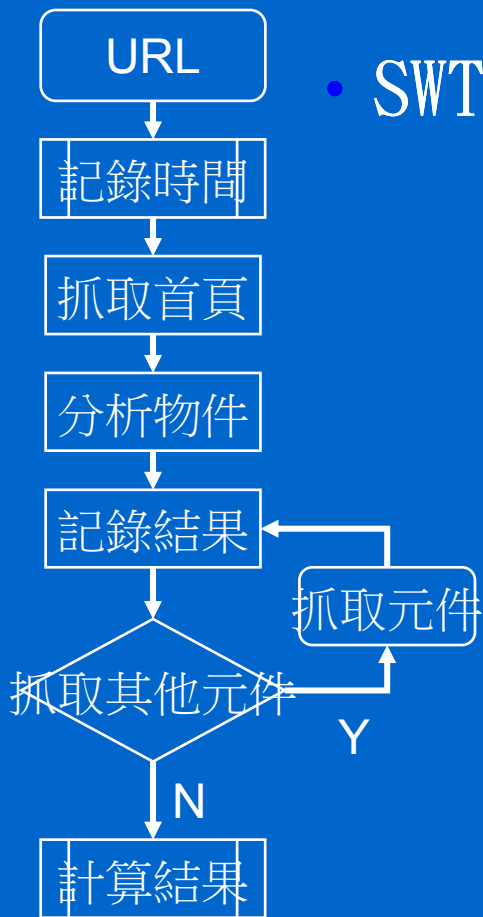
- 優點

- 運作於傳輸層 (Transport Layer)
 - 利用開啟 TCP socket 方式偵測網路設備或伺服器連線狀態，可針對所有的 TCP 網路服務進行測試
 - 可避免機器本身連線正常，但無法提供服務的情形
 - 通用性高，不必對每種服務撰寫專屬程式

- 缺點

- 仍有可能發生開啟 TCP socket 正常，但伺服器服務異常
Ex: 資料庫資料表被鎖住

相關的網路品質狀態量測方法



• SWTM(Simple Web Traffic Meter)

- 基於應用層的網路品質量測系統
- 量測使用者從連接網站至首頁完全呈現所需時間
- 可連線到指定的網站判讀HTTP的狀態碼
(ex:[200 OK]、[404 Not Found])

- 自動分析網站內容，記錄取回網頁及所需元件的傳輸時間

• 缺點

- 僅針對WWW網站首頁呈現時間量測，並無對網站的其他相關服務做品質量測
- 依然可能發生Web server本身量測結果正常，但無法對資料庫進行資料存取的情況

系統架構流程與設計

- cURL可以
 - 模擬使用者與網頁伺服器做各種自動化互動並記錄花費時間
 - 偵測出Layer3正常但Layer7異常的情況(因為它運作於應用層)
 - 針對網頁伺服器上特定服務或功能進行服務可用性偵測
- cURL是
 - 利用URL語法在命令列下運作的傳輸工具
 - 支援傳送和接收的綜合傳輸工具
 - 支援FTP, FTPS, HTTP, HTTPS, SCP, SFTP, TFTP, TELNET, DICT, LDAP, LDAPS and FILE 等通訊協定
 - 支援SSL certificates, HTTP POST, HTTP PUT, FTP uploading, HTTP form based upload, proxies, cookies, user+password authentication 等模擬方法

系統架構流程與設計

- 採用PHP/cURL script方式
 - PHP不僅可以撰寫網頁程式亦可以script方式在背景執行
 - 當取得回傳http code或網址等資料後，易於進一步分析處理
 - PHP/cURL有完整的參考資料與範例程式
<http://tw.php.net/curl>

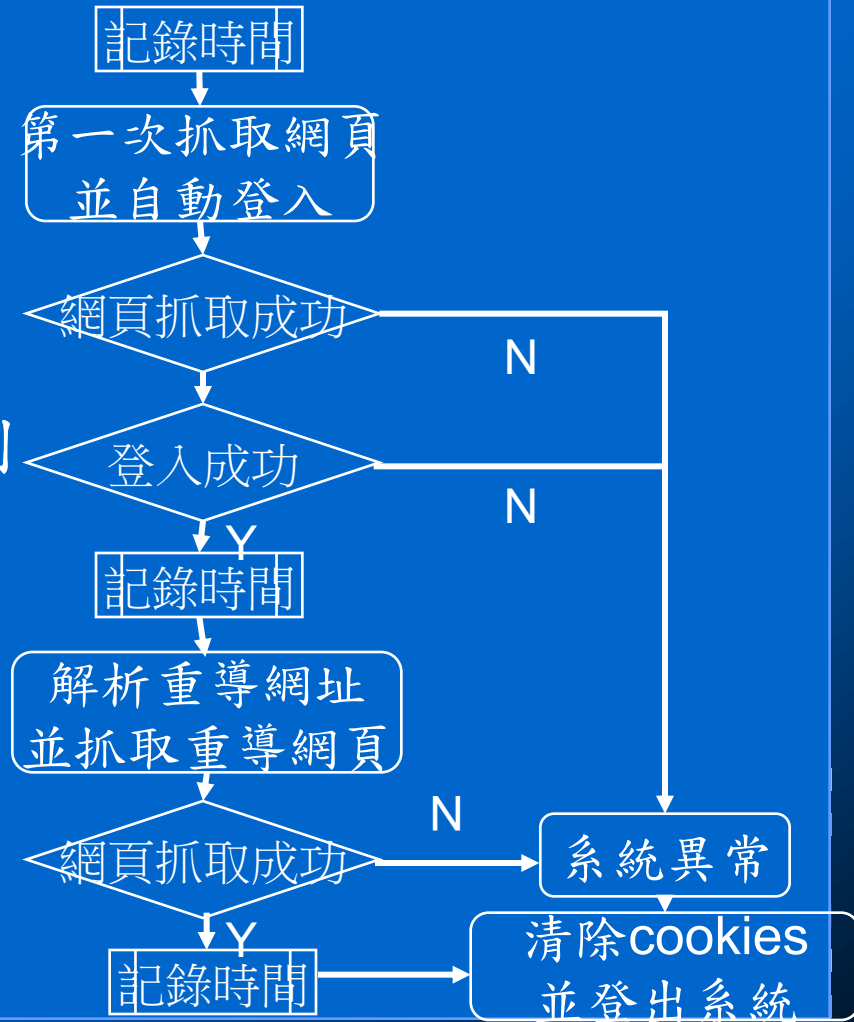
系統架構流程與設計

- PHP/cURL自動化登入量測程式設計
 - 針對本校moodle教學平臺入口網站做自動化登入
 - 記錄自開始載入登入頁面，至登入成功並完全載入重導網頁頁面所耗費時間
 - 達到偵測網頁伺服器服務狀態目的
 - 分析登入moodle教學平台的流程與架構
 - 先利用獨立的cURL.exe程式搭配適當參數，對moodle入口網頁登入流程進行測試與分析
 - 再將測試分析結果透過PHP/cURL改寫成script在背景定時執行

系統架構流程與設計

- PHP/cURL 自動化登入量測程式流程

- 若回應網頁不正確或等待時間超過timeout值，則判定為**登入失敗**



系統架構流程與設計

- php/cURL 抓取登入首頁程式片段

```
2 <?php
3
4 function L7_ping($url, $timeout) {
5
6     //每次偵測都隨機產生一個cookie檔
7     $cookie = mt_rand()."cookie";
8     $total_time = $timeout;
9     $dynamic_timeout = $timeout;
10
11     //準備抓網頁
12     $ch = curl_init($url);
13     curl_setopt($ch, CURLOPT_POST, 1);
14     curl_setopt($ch, CURLOPT_POSTFIELDS, "foo=test&bar=test2");
15     curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, FALSE);
16     curl_setopt($ch, CURLOPT_RETURNTRANSFER, TRUE);
17     curl_setopt($ch, CURLOPT_COOKIEJAR, $cookie);
18     curl_setopt($ch, CURLOPT_COOKIEFILE, $cookie);
19     curl_setopt($ch, CURLOPT_HEADER, 1);
20     curl_setopt($ch, CURLOPT_TIMEOUT, $dynamic_timeout);
21
22     //計時開始
23     list($start_usec, $start_sec) = explode(" ", microtime());
24     $start_time = ((float) $start_usec + (float) $start_sec);
```

系統架構流程與設計

- php/cURL 判斷登入是否成功程式片段

```
39 //如果成功, http header 會是 303, 且內含 redirect 的目的地
40 if ($output != false && $info['http_code'] == 303 && $total_time < $timeout) {
41
42     //解析要 redirect 到哪個 url
43     $begin_tag = "location.replace('";
44     $begin_position= strpos($output, $begin_tag) + strlen($begin_tag);
45     $end_tag = "');";
46     $end_position = strpos($output, "');");
47     $length = $end_position - $begin_position;
48     $url = substr($output, $begin_position, $length)."\n";
```

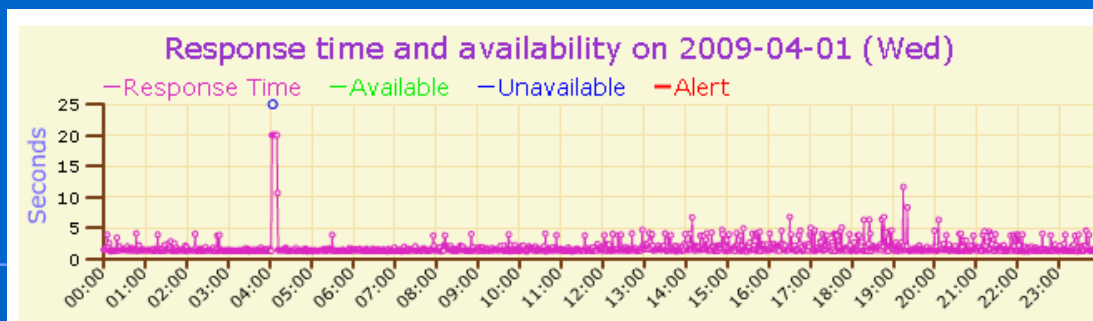
系統架構流程與設計

- php/cURL 登入成功抓取重導網頁程式片段

```
50 //準備抓下一個網頁
51 $ch = curl_init($url);
52 curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, FALSE);
53 curl_setopt($ch, CURLOPT_RETURNTRANSFER, TRUE);
54 curl_setopt($ch, CURLOPT_COOKIEJAR, $cookie);
55 curl_setopt($ch, CURLOPT_COOKIEFILE, $cookie);
56 curl_setopt($ch, CURLOPT_HEADER, 1);
57 curl_setopt($ch, CURLOPT_TIMEOUT, $dynamic_timeout);
58
59 //抓網頁 -- 第二次 -- 進入登入成功後的第一個頁面
60 $output = curl_exec($ch);
61 $info = curl_getinfo($ch);
62 curl_close($ch);
63
64 //結算到目前共花了多少時間
65 list($end_usec, $end_sec) = explode(" ", microtime());
66 $end_time = ((float) $end_usec + (float) $end_sec);
67 $total_time = round($end_time - $start_time, 2);
68 $dynamic_timeout = ceil($timeout - $total_time);
69
70 //如果成功，http header 會是 200，且內含登出的 link
71 if($output != false && $info['http_code'] == 200 && $total_time < $timeout
72     && (strpos($output, "http://test.url/logoff.php?session_key=") != false)) {
```

量測結果與結論

- PHP/cURL 實際模擬一個client向伺服器要求服務
 - 不僅僅量測網頁載入時間
 - 一次量測流程，可同時針對web server、網頁程式、後端DB server進行測試
 - 真正反應使用者實際使用系統服務的狀況



Zoom [x2](#) [x4](#)

量測結果與結論

- 與其他量測方法相較之優點
 - 彌補單靠SNMP取得資訊(ex:CPU使用率、記憶體使用率、網路頻寬使用率)之不足處
 - 沒有安全性上的問題，故不會有類似ICMP封包被阻絕的問題
 - 可以解決利用開啟TCP socket連線偵測設備正常上線方式，仍無法保證網站服務正常的問題

未來工作

- 本文提出的機制，可依照需求不同，對網頁伺服器各項服務功能進行偵測
- 計劃發展一套能夠判別系統是否真正失效的斷線回報機制
 - 結合PHP/cURL量測的數據，與SNMP所取得的網路設備或伺服器資訊
 - 找出一合適的Model用於判別失效警報發送與否的機制，減少不必要的失效警報發送



The End

