# TWAREN 網管 Flow 監控機制之設計與實作

梁明章
國家實驗研究院國家高速網路與計算中心
liangmc@narlabs.org.tw

## 摘要

本文將說明 TWAREN 在 FLOW 資料處理方面的目前已有的成果，包含常規統計分析與報表，特定條件的線上即時查詢機制，並說明嘗試以分散運算方法進行較為隨意條件的線上即時查詢等研究方向。

**關鍵詞**：TWAREN，FLOW，雲端運算。

## 1. 前言

TWAREN 網路屬於光電混合網路，設備含括多層，而學研網路的研究使命使得 TWAREN 網路必須時常換代提升讓國內學研界有接觸先進網路技術的機會，為節約國家經費，TWAREN 每次換代並非全面更新打造另一套骨幹，而僅是加入新技術設備，與前數代設備共同運作，幾次下來，TWAREN 網路越形複雜，因此數年前維運團隊就開始自行研發網路管理系統，主要目的就是期望能親自掌握網路管理的核心技術，才能因應頻繁更新的 TWAREN 網路架構與設備，使網路系統亦能隨之擴展成長，而不需要置辦越來越多套業界軟體或甚至全體撤換。

我們針對網管系統的研發主要分下列幾個方向：

**狀態異常監測**：主要工作就是「異常偵測」，想讓電腦系統判斷「何謂異常」，就必須告訴電腦「比較值」或稱「基準值」，狀態主要分兩類，第一類是二分狀態，例如線路通或斷(Up/Down)，這類判斷比較簡單，第二類則是量化數值，例如流量、溫度、電壓等等，這類判斷就必須定出基準數值，單一基準數，或是兩個基準數(上下水標線)，或甚至是多重基準數(多等級水標線)，因此網管系統對此類異常偵測的靈敏度與正確率，全看基準值是否適合，為了取得合適的基準值，我們將多種狀態數值長期記錄並透過類標準差算法每五至三十分鐘自動推測下次比對所需的基準值，而非使用人工設定固定數值的方法，針對一些波動較大的偵測標的，我們還加上緩衝判斷頻率機制，使得偵測靈敏度提升之際，也不至於損害太多準確率，這個動態臨界基準值計算我們曾發表於 TANet2009[1]。

**品質效能監測**：這類標的的異常判斷極為模糊，有些甚至是見仁見智，除利用動態臨界基準值計算之外，還需參考使用者感受進行調整。此外，我們發現藉由品質數據的長期變化趨勢，可能推測出品質劣化即將產生障礙，或是惡化趨勢加快可能表示劇烈波動(劇烈波動可視為異常)，因此我們也將此趨勢偵測預測方法應用至網管系統當中，相關成果也發表於 TANet2009[1]。

**使用分析與查詢**：傳統網路使用分析大部分指頻寬應用在哪些方面，通常是利用 Flow 資料進行分析與統計，常見應用是排序使用者用量 TopN，以及將每五分鐘的平均值送入 MRTG 或 RRD-Tools 做圖表呈現，站在骨幹維運者角度來看，由於 Flow 資料增量速度極快，若要即時監控，需要相當大的計算力才可能追上。我們在這幾年持續進行 Flow 相關的研發，先期開發可預先背景處理的分析與統計，近期則逐漸涉入 BigData 分散計算，嘗試網頁即時隨選查詢是否能實現。

**自動控制設定變更**：我們曾開發一些自動變更簡單設定的功能，例如當網管系統偵測到線路介面斷線時，自動登入該設備對該介面下達重啟指令，有時候故障會因此排除，無需等待工程師上線來處理。然而要自動進行更複雜的操作卻有極大的風險，因為透過網路連線下指令本身就有斷線的風險，若由工程師操作，萬一半途不幸斷線，以工程師的人類智慧可以應變並補救，然而程式要自動應變那麼多種可能性實在很困難，因此這部份我們暫時就不再深入開發了。

本文將把重點放在 Flow 相關的網管開發，說明我們在這方面已有的成果與未來的開發設計。

## 2. 設計與方法

我們將 Flow 的處理方式依照即時與非即時區分成三種：

### 2.1 完全預先處理

一般常規報表屬於這一類，例如每天輸出一次使用者用量排行表、各類應用排行表、連線單位用量排行表，各種 MRTG 與 RRD-Tools 圖表等等，這類事先制定的運算可以在背景定期執行，只要程式

能在定時周期內完成計算，不會拖累到下次的計算即可，這類程式最可能面臨的瓶頸在於讀取資料這一關，由於 Flow 資料量龐大，如果只存一份，要同時進行多組計算，資料讀取會形成瓶頸，解決的方法基本上就是用錢買空間來換取時間縮短，例如多買硬碟空間多存幾份，像 Hadoop 的 HDFS 就是同樣的資料多存幾份來協助加速，複製份數越多，加速就越明顯。

## 2.2 部分預先處理+部分即時運算

當我們開發好前述完全預處理機制後，有時候會需要查詢某特定時區間內的統計與分析，例如指定某日至某日之間或某時刻到某時刻之間的合計，這種可以事先規範好的查詢條件，可以事先做好前期資料準備，好在查詢條件確定後，能利用這些事先做好的部分資料快速完成計算。

由於近年來使用者習慣利用瀏覽器操作網頁程式，因此這種查詢會面對瀏覽器的接收逾時問題，一般預設是 15 秒，亦即網頁資料如果無法在 15 秒內傳輸完畢，有可能會被瀏覽器當作逾時而中斷，下面將說明我們為了這關鍵 15 秒做了哪些處理。

網管系統每天按照跨洋電路、Internet 互聯電路、連線單位等等個別進行用戶 IP 的用量統計、Port 應用量統計，以及目標地所屬 ASN (Autonomous System Number) 用量統計，下**表 1** 是一些統計數據，表示我們每天面對的 IP 用戶數量，當然這些 IP 日常就包含了不少 Fake-IP 的封包，所以像中山大學的 IP 總量雖然只有一個 Class-B，卻依然每天都統計到七萬多個 IP 用戶，而高屏澎區網在 6 月 3 日更是狂飆到平日的四倍多，也是超過分配 IP 總量，表示當日有比較多的 Fake-IP 疑似攻擊出現。

**表 1 統計範圍與用戶 IP 數量舉例表**

| | 歐美研網 | 亞太研網 | 高屏澎區網 | 中山大學 |
|---|---|---|---|---|
| 2013/6/1 | 5331694 | 956774 | 773551 | 70456 |
| 2013/6/2 | 5339895 | 971099 | 769770 | 70591 |
| 2013/6/3 | 5352203 | 939221 | 3224229 | 70786 |

我們用前述的用戶 IP 用量統計、Port 應用量統計，以及 ASN 用量統計按照每個 IP、PORT、ASN 做關鍵值進行計算，存成定時結算列表，目前儲存日結算表與月結算表，未來若儲存空間足夠的話，預計增加每小時結算表，甚至每十分鐘結算表，這些資料足以進行精確計算，然而，對於在網頁上進行即時查詢的功能來說，查詢者通常不會關心到每一個 IP 或每一個 PORT 的用量，一般查詢者只會關心前幾名，亦即排序後的結果。

數十萬或百萬筆資料的即時排序要在 15 秒內完成並不容易，因此我們必須先排序好，並儲存排序結果，然而考慮到節約儲存空間，我們放棄儲存完整的排序結果，從經驗來看，網路大部分的資源都被少數人用掉的，而下**表 2** 與下**圖 1** 也證實了這個經驗，若只儲存用量超過平均值的名次，則我們僅需儲存前面百分之五不到的名次資料，節省許多儲存空間。

**表 2 用量高於平均值的 IP 用戶占比表**

| IP 占比 | 歐美研網 | 亞太研網 | 中山區網 | 中山大學 |
|---|---|---|---|---|
| 8 月 14 日 | 0.26% | 1.74% | 3.69% | 5.08% |
| 8 月 15 日 | 0.41% | 1.97% | 0.44% | 4.34% |

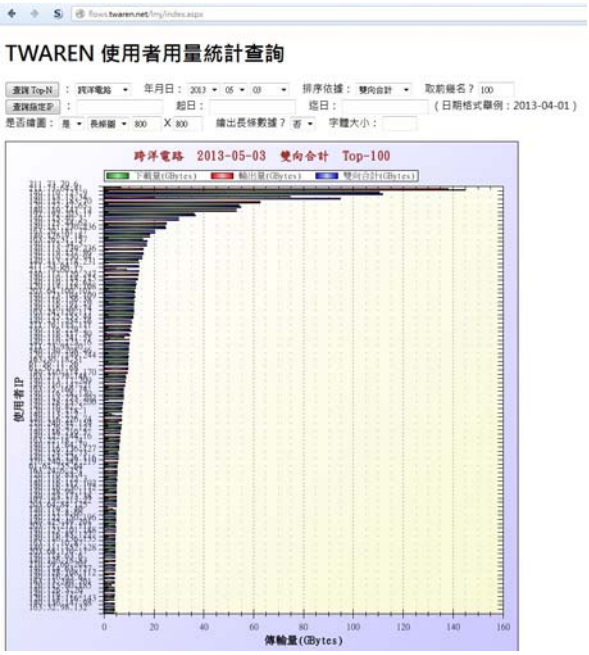


**圖 1 頻寬用量集中於少數 IP 大戶的現象**

預先處理的儲存結果，其資料大小遠小於 Flow 原始資料，因此在線上即時查詢時，所需消耗的讀取時間也大量減少，也能整批讀進記憶體中再運算，盡量減少等待硬碟的時間，使得一些線上即時查詢能進入關鍵 15 秒內，例如下**圖 2**，就可以線上查詢某特定 IP 在一個半月內的日用量而不會逾時，其實可以查詢更長的區間，只是圖表會拉得太長佔據太多篇幅，所以在此不舉例。

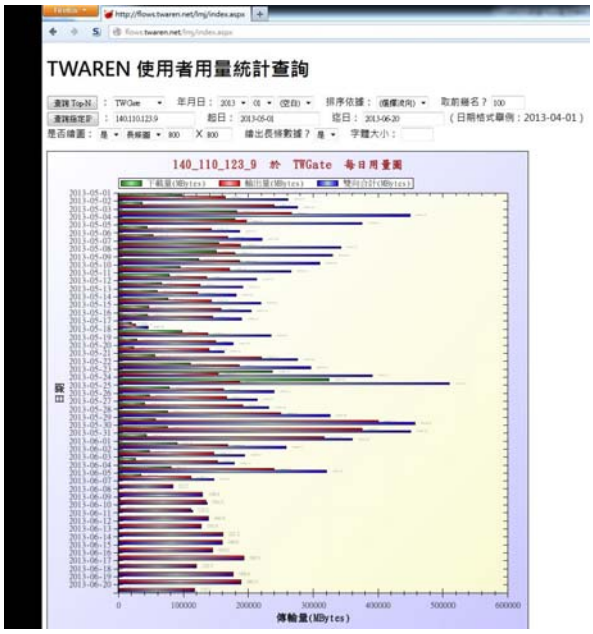


**圖 2 特定 IP 用量查詢**

目前我們仍持續研發更多利用局部預先處理方法來協助特定的線上查詢，而不會將全部希望都放在分散式雲端運算平台上，因為對我們來說雲端平台是蠻燒經費的一個選擇。

### 2.3 即時運算

儘管分散計算平台非常燒錢，然而不可否認的是，他確實是最有可能實現大部分線上即時查詢功能的方向，我們拜讀了一篇 IEEE Workshops 的論文[2]，該文探討透過 Hadoop 平台利用 Map-Reduce 方法運算 PORT 應用量統計，並量測時間比照，擷取該論文一張圖表如下**圖 3**，可以發現 Map-Reduce 的 data-node 逐步增加之後，其縮短的時間比率逐漸不如 node 數量之預期，以 TWAREN 跨洋電路一天的 Flow 筆數 300 個 million 來估計，Flow-Tools 約需 10 小時，按下圖比例若想要縮短到 15 秒內，可能需要出動 Google 資料中心那樣的貨櫃車。

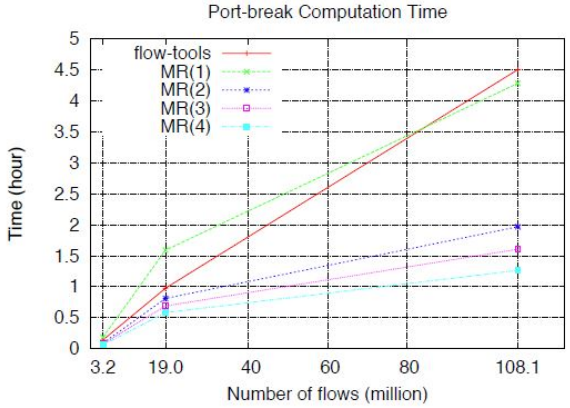


**圖 3 Flow-Tools 與 Map-Reduce 耗時比較**

因此，我們目前在 Hadoop 上的研究重點暫時不放在即時運算上，而是先將目標放在多種類的統計與分析上，例如，過往我們若想做一天的結算，譬如 IP 的用量排行，一天的資料由單機跑往往需要數小時，倒不是 CPU 太慢，而是因為讀取資料量的筆數迴圈太多次(每筆一次 I/O 要求是相當花費時間的)，而這一個問題若交給 HBase 就能比較快速解決，但問題是，大量的 Flow 資料要放進 HBase 卻需要相當多的時間，對於每日都需要計算新資料的情況來說，其實也要耗費不少時間，不過幸運的是，資料只要花時間進了 HBase，那麼後續不論做幾次不同的統計分析，都可以運用 Map-Reduce 來加速，而不用像傳統方法那樣每做一種不同的統計分析都得重新讀一遍 Raw Data 那麼久。

## 3. 未來工作

我們仍將繼續雲端分散計算的方向研究，畢竟他是目前最可能達成即時運算的方向，若輔以相當的預先處理，或許可以達成我們的目標，我們將先研究 Hadoop 以及 HBase 與 Flow 資料結合的方法，以及與之輔助的預先處理方式，希望能找到可以較少的 Data-Node 發揮足夠運算力的路。此外，Flow 資料常常是一個 Key (如 IP) 帶至少兩個 Value (In/Out 雙向流量) 甚至多重結構 (例如一個 IP 有 65536 個 PORT，每 PORT 各有 In/Out 雙向流量)，在做比較複雜的統計分析時，尤其是同時須對應多方 IP 與多 PORT 關聯時，要如何組合多個 Key 與值以符合 Map-Reduce 的 Key-Value pair，這也是我們目前研究的課題之一。

## 參考文獻

[1] 梁明章, 張聖翊, 林孟璋, 謝欣叡, “TWAREN 混合型網路管理系統之進階異常偵測與拓蹼監控技術,” TANet2009, 2009.

[2] Youngseok Lee, Wonchul Kang and Hyeongu Son, “An Internet Traffic Analysis Method with MapReduce,” 2010 IEEE/IFIP Network Operations and Management Symposium Workshops, 2010.