TWAREN 骨幹分散自治型網管架構設計

梁明章　陳俊傑
國家高速網路與計算中心
liangmc@narlabs.org.tw　jjchen@narlabs.org.tw

摘要

本文將說明 TWAREN 現行的網管系統架構與面臨的問題，數千上萬個監控項目在一輪(五分鐘)時間內難以如期完成一輪工作，說明效能瓶頸所在，以及現行架構執行智慧化判斷的困難，最後針對上述問題提出分散自治型態的網管架構設計，本文將說明架構內容以及解決問題的方法。

關鍵詞：TWAREN、網管、智慧化、AIOps、大資料、分散自治。

1. 前言

台灣高品質學術研究網路 (TaiWan Advanced Research and Education Network, TWAREN)**[1]**在初始創建時，光設備與路由設備各自附帶獨立的原廠管理軟體，為了整合網管，還加了幾個前置與後置處理軟體，四五種企業級軟體東接西串共同運作，建置時為了各軟體間資料流的轉換串接傷透了腦筋，在實際運行後，障礙發生時常在某些環節卡住導致最後沒形成告警，除錯很困難，因此 TWAREN NOC 維運團隊痛下決心，自行研發網管系統，目標只是為了符合 TWAREN 設備的監控告警需求，以及事件追蹤管理，以 NOC 實用為主，追求「不漏警、不誤警」，由於維運團隊成員都是網路工程師，開發過程也是不斷地學習與試錯，架構推倒重建數次，由於沒有程式設計專業，善用語言也不統一，為了方便分頭開發時的互相搭配，TWAREN 網管系統從誕生之始就不是一個單獨的程式，而是以一個核心資料庫為中心，眾多程式做為衛星環繞核心運行的架構。

我們以免費開源的 MySQL 作核心資料庫，近幾年更換成免費開源的 MariaDB Cluster 加 HA-Proxy，以分散在兩地機房的異地備援方式運行，資料表分類如下**表1**：

表 1 核心資料表分類簡述

大分類	小分類	用途與說明
設備屬性類	屬性主表	指定設備 ID 主鍵，關聯機房 ID 主鍵，必要資訊，各監控項目臨界基準值與狀態正常與否，是否監控等等
	CPU、記憶體、電壓電流、風扇、溫度等設備相關監控項目	各監控項目表都關聯設備 ID，並分成最新狀態數值表與歷史數值表，每個數值都附帶取得時刻
機房與連線單位類	屬性主表	分為機房與連線單位兩個主表，指定機房 ID 與連線單位 ID 主鍵，連線單位會關聯機房 ID，記載必要屬性資訊，聯絡人，經緯度，單位路由表，各監控項目臨界基準值與狀態正常與否，是否監控等等
	環境溫度、不斷電系統、單位流量、單位路由、傳輸品質、單位內用量 Top10 IP&Port 等等監控項目	各監控項目表都關聯機房與單位 ID，並分成最新狀態數值表與歷史數值表，每個數值都附帶取得時刻，單位路由每次都進行精確比對
線路與介面類	屬性主表	定義實體線路/虛擬線路/光路徑的 ID 主鍵，關聯設備 ID，記載線路屬性資料，監控方式，臨界基準值，目前狀態正常與否，是否監控等等
	傳輸狀態、傳輸品質、光功率等等監控項目	各監控項目表都關聯線路與介面 ID，並分成最新狀態數值表與歷史數值表，每個數值都附帶取得時刻
Peering 類	屬性主表	定義網際互聯 ID 主鍵，記載對端單位資訊，關聯設備 ID 與介面 ID，雙方交換路由筆數上下臨界值，狀態正常與否，是否監控等等
	狀態、雙方路由筆數、傳輸品質等監控項目	各監控項目表都關聯 Peering 主表 ID，並分成最新狀態數值表與歷史數值表，每個數值都附帶取得時刻
Layer2 VPN 類	VPN 服務主表	定義 Layer2 多點 VPN 服務的 ID 主鍵，記載用戶聯絡資訊，VPN 狀態正常與否，是否監控等等
	VPN 節點表	定義每個 VPN 節點資訊，關聯 VPN 服務 ID 與介面 ID，節點狀態正

		常與否
	狀態、傳輸品質等監控項目	各監控項目表都關聯 VPN 節點 ID，並分成最新狀態數值表與歷史數值表，每個數值都附帶取得時刻
事件類	異常事件表	紀錄異常的監控項目、異常類別、嚴重性等級、開始異常的時刻、異常結束的時刻
	事件單追蹤表	紀錄開出到國網中心事件單系統的單子、事件單類別與等級、關聯有關的異常事件、開始時刻與結案時刻
特殊類	網際網路服務、臨時專案	網際網路服務伺服器相關監控、特殊監控或通報流程等等無法納入常態監控程式的項目，以特別的表與特別修改的程式來處理，以免導致網管程式過於複雜

從上表可以看出我們刻意將資料表拆分為主屬性表、最新狀態數值表、歷史數值表，這與我們將網管系統拆分成很多小程式各自運行有關係，為了將修改瓶頸壓力分攤，後文會再說明。

因為我們不可能花上幾年時間開發一個大系統直到完成才來使用，我們是邊研究網管技術邊增加監控項目納入網管系統的，從一無所有到粗略監控，逐步完善到細密監控，我們走了很多年，為了便於工程師個別開發、測試，以及隨時增加功能，我們決定以許多小程式來組成網管系統，程式群主要分成如下**表2**所述幾個階段群：

表 2 網管程式群分類簡述

階段	大分類	小分類	說明
資料獲取	主動抓取	MIB Polling	從核心資料庫取得設備 SNMP 資訊並根據監控標的 OID 取得數值，修改最新狀態數值表，並新增至歷史資料表。 由於各設備的 SNMP 實現程度不一，Private MIB 更是各異其趣，甚至設備更新 OS 都可能造成 SNMP 實現改變，設備大換代更是得大改特改，為了避免程式內因為設備差異充斥大量 If/Else 判斷不好維護，這部分程式群是被拆分得最瑣碎的部分
		SSH Telnet HTTP RESTful	無法從 SNMP 獲取的資訊，則以登入系統下指令或 RESTful Get 然後分析回應內容的方式取得數值，基本都靠字串分析，因此常因設備 OS 升版或 Patch 導致字串略有差異而出錯，也是極難維護的部分
	被動接收	Trap SysLog Netflow	Trap 與 Syslog 格式多樣多變，分析規則必須靠時間積累才能逐漸完善，而且常因設備 OS 升版或 Patch 導致字串略有差異而出錯，極難維護。 Netflow 則是需要龐大運算力，成本甚高。
異常判斷、告警與開單	狀態判斷	介面狀態、連線狀態等等	比對值是有限數字集合，例如0/1(通/斷)，或是錯誤代碼之類的，最容易判斷
	數值判斷	CPU/記憶體/溫度/電壓/風扇轉速/流量等等	非有限數字集合，必須在資料庫指定臨界值，很多監控項目無法共用臨界值，有些需要上下水標臨界值，有些需要多重臨界值以區分嚴重程度，諸多臨界值的設定與維護是個大難題
	品質判斷	RTT/Packet-Lost/網際網路服務響應時間/流量滿載百分比	沒有異常狀態發生不代表正常與滿意，為了貼近使用者的觀點，我們也實現了幾項品質監控，未來仍有改善空間
	行為判斷	過量使用/惡意行為/癱瘓行為	網管平時就需要遏制與警告會造成全體困擾的過量或癱瘓行為，以避免有急需時來不及處理。判斷惡意行為是為了降低使用者的資安風險。我們利用 netflow 的即時統計與分析來判斷行為。
	分級判斷	嚴重性等級判斷	根據預先定義的嚴重性規則自動區分異常為「僅記錄」「UI 介面黃色警訊」「UI 介面紅色告警與電話簡訊並開單」，因為目前智慧程度仍不足，有些較難

			自動分級的異常仍由 NOC 值班一線判定。
事件追蹤	異常項目	異常追蹤	監控項目偵測到異常，就會新增記錄到異常事件表，並週期性追蹤，直到該監控項目恢復正常或被標註暫停監控，才會寫入該筆異常的結束時刻
	事件單	事件單追蹤	異常事件發生時，根據嚴重性規則判斷後如需要開立事件單，則新增到事件單追蹤表並透過開單 API 對國網中心統合事件單系統開單，並週期性查詢事件單系統是否有結案關單，如已結案，則在追蹤表標註該單的結案時刻
使用者介面		整合式監控平台 WEB	UI 平台週期性到核心資料庫讀取各種資料進行畫面呈現，是 NOC 值班一線監看的畫面，也是值班二線處理事件時的查詢參考
		大資料即時查詢平台 WEB	這幾年建置大資料平台提供強大的即時運算能力，使 NOC 可以即時監控異常行為、即時告警進行遏制

簡單描述 TWAREN 網管系統的運行階段，就是先使用資料獲取程式群將取得資料處理後寫入核心資料庫，然後執行異常判斷程式群從資料庫取出需要的資料，根據各種規則判定是否異常、告警、開單，再將判斷結果寫回資料庫。同時追蹤程式群也必須週期性讀取資料庫以判斷異常是否消失、事件單是否結案。而使用者介面(UI)也要週期性讀取資料庫呈現監控畫面。以上所有程式幾乎都是以每五分鐘為一個輪迴週期來運行，UI 更短，每分鐘一輪迴。

雖然開發過程歷經許多困難，但 NOC 團隊也因此掌握網管系統的核心技術，不受制於設備原廠的網管軟體，後來 TWAREN 經歷幾次設備與線路架構的大換代，我們的網管系統總能跟著調整適應，不會面臨換設備就得換一套網管的困擾。

TWAREN 網管第一目標是追求「零漏警」，自然會陷入告警越來越多的窘境，因此加入第二目標「有效率的告警」，也就是告警的分類與分級技術，在多年前我們就曾在 TANet2009研討會提出論文「TWAREN 混合型網路管理系統之進階異常偵測與拓蹼監控技術」**[2]**，以歷史資料動態演算下一次臨界值的合理範圍，並且利用趨勢演算來緩衝可能過於敏感的合理範圍所增加的告警，並利用趨勢來輔助判斷嚴重等級，只可惜當年 NOC 的演算資源不足，只能實作在幾個重要監控項目上。

隨著 TWAREN 換代升級，設備與種類漸多，為了提升備援能力，架構與拓樸越趨複雜，即使發生十幾個告警也不見得會影響網路服務，因此告警分級與嚴重性判斷日趨困難，我們越來越感覺到網管的智慧化需求，本文將說明 TWAREN 網管面臨的重要問題，以及我們構思的解決方案。

主要內容

前言提到 TWAREN 網管十多年來邊學邊開發，一路發現問題並解決問題，雖然開發成員是網管工程師兼任，不是專業程式師，寫不出美美的偉大系統，但一路走來也累積了不少網管開發技術，但是因為這幾年免費大資料演算平台軟體普及，NOC 運算力大增，加上 AIOps 的風向，我們也終於面臨到現行架構難以解決的問題，接下來我們將主要說明兩大問題以及解決方案。

1.1 現行架構面臨的問題

開發十多年來，系統監控的項目與標的，隨著時間發展越來越多，如前言所述，為了不漏掉任何異常，監控項目從實體介面/線路/設備元件/機房環境，逐步延伸到邏輯介面(VLAN/Trunk/Bundle 等等)、多點 VPN、邏輯專線，再加入多種品質監控，數千上萬種監控項目，大部分資料都需要向設備索取，頻繁的 SNMP MIB Polling 與 SSH 登入，對設備形成壓力，眾所周知，網路設備的處理器設計是優先處理傳輸與路由演算，對網管需求的優先級並不高，當網管系統索取過度時，回應會變慢甚至逾時，累積結果就是五分鐘該執行一輪的資料收集無法在五分鐘內完成，甚至會發生後面兩三輪的收集工作一起堆積，後輪搶先完成的錯亂現象使監控資料失真嚴重，發生資料空窗等多種問題。

我們試過將循序工作改成多行程併發背景執行，結果瓶頸改到核心資料庫，競爭資料表鎖定的等待隊列拉得太長，同樣造成逾時，寫入資料逾時的後果比取資料逾時還難處理，而且併發對於設備造成的壓力更大，取資料逾時現象更嚴重，併發執行還有一個嚴重後果，就是很難掌握資料獲取階段完成與否，影響異常判斷階段不知何時才能啟動，得此失彼，結果更糟。

最後我們使用折衷方式，就是盡量將資料獲取與異常判斷之間資料有必要順序關聯的程式合併成群，同一群內的程式依序執行確保寫入資料庫的先後，各群則併發執行，勉強降低同一輪內先後錯亂的現象，對於併發帶來的逾時影響到不同輪迴的先後錯亂現象也有一些改善，但依然無法完全避免，我們雖然努力改善資料庫效能，但是對於資料來源的設備網管效能仍是無可奈何，以上就是因為監控項目過多帶來的議題。

第二個議題就是智慧化判斷，監控項目越多，出現告警的頻率就會隨之增高，但是因為骨幹網路拓樸架構的備援性設計越來越好，可繞行的路徑很多，即使併發多個告警也未必影響服務，雖然終歸要處理，但不一定得要半夜擾人清夢，因此自動智慧化分類分級更加重要，此時我們發現，已經使用十多年的核心資料庫不太適用於智慧化演算，因為核心資料庫是 SQL 關聯式資料庫，平常網管運行頻繁的新增與修改就會造成頻繁的鎖定排隊，使得讀取效率也下降，智慧化演算常需要讀取大量資料作綜合判斷，受到影響，大概五分鐘內也無法完成一輪，因此我們需要思考另闢途徑的方法，因為智慧化判斷除了可以提升告警效率之外，也可以用來判斷問題根源(根因分析)，提供給值班工程師做為排除障礙的重要參考。

本文將針對上述兩個問題提出一個網管程式架構來嘗試解決，以下說明我們的設計方法，並且加入 ElasticSearch Cluster 大資料平台(後文簡稱 ES)作資料庫之一。

1.2 分散自治型網管架構設計

近年來普及的大資料平台給了我們很好的設計思路，TWAREN 網管系統原本是以程式讀取核心資料庫得到有哪些設備的哪些項目要讀取資料，然後一台台的讀過去，所有程式由系統的 CRON 排程服務照週期啟動，如同以中央的角色定期去各地方巡察，我們轉換思考，何不改成地方自治，有事時才向中央回報？於是我們想到分散自治型架構，用物件導向語言開發，整體架構如下**圖1**所示：

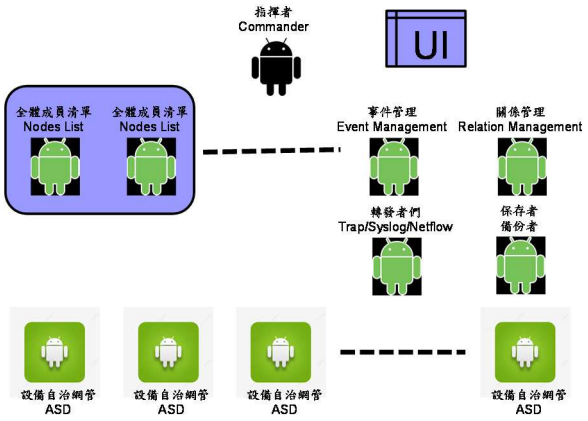

圖 1 分散自治型網管系統架構圖

各角色分別說明如下，所有成員之間的溝通都透過網路，所以無須執行於同台機器或是同地機房：

◎成員清單維護者(Nodes List Daemon，簡稱 NLD)：乃是本系統必須最優先存在的 Daemon，負責記載所有成員執行體目前所在的 IP 位址與 Port 清單，預計以一個 Primary 加一至二個 Secondary 存在，Primary 隨時同步清單更新給 Secondary，所有成員一啟動就必須向 NLD 登記註冊，並從 NLD 獲得其他成員資訊，NLD 也負責檢查各成員存活狀況。

◎自治網管(Autonomous System Daemon，簡稱 ASD)：以一個常駐母行程對應一個設備，自主運行，每個 ASD 初次啟動先讀取設備資料或設定檔在記憶體中建置虛擬設備物件，以設備為物件主體，掛上各種監控項目物件，各監控項目物件可以子行程或執行緒各自運行，監控項目物件屬性紀錄項目數值、狀態、臨界值(若需要動態臨界值就自己算)、取得時刻等等，其物件方法負責自身的屬性值如何取得，取得後隨即可判斷是否符合臨界值，若有異常則向設備母體物件行程通報，母體物件轉通報給事件管理者，同時記錄到 ES 的設備紀錄區。母體物件同時也接收轉發者轉來的該設備 Trap/Syslog/Netflow 等即時資訊立即分析，如有需要，可以觸發相應監控項目物件立即向設備重新取得數值作確認及補充資訊，若從 Syslog 收到 Configuration 組態設定變更的訊息，就用 SSH 或 SNMP 去取得新組態設定並觸發自身重構。母體物件也負責定期將體內當下資訊儲存到 ES。如果某個 ASD 需要遷移到其他機器或重啟，可以將復原所需的資訊轉換成 JSON 文件先送到備份者那邊儲存，在遷移或重啟後取回，可以快速回復到先前狀態。

◎關係管理者(簡稱 RMD)：負責調查並維護各 ASD 與其他 ASD 的關聯資訊提供查詢，例如透過相同子網段、相同 VLAN、路由 Neighbor 等等建立 ASD 之間的介面關聯、監控項目關聯，對於異常是否影響哪些服務的綜合判斷，以及異常根源的智慧判斷可以提供相當有用的輔助資訊。當有 ASD 發生重啟或重構時會通知 RMD，RMD 會在延遲一點時間後(為了判斷是否為大量重啟)重新調查關聯。

◎事件管理者(簡稱 EMD)：接收各 ASD 傳來的異常通報，收到後等待一小段時間(時長尚待研究，為了等待其他有關的 ASD 通報)，向 RMD 查詢各通報的所屬關係群，關係群重疊的通報極可能是相關通報，彙整分析結果存入 ES 中，並判斷是否需告警或開單，EMD 存入 ES 的資料對於異常根源智慧化分析演算會很有幫助，是 AIOps 重要資料來源。至於後續追蹤工作可能由 EMD 處理也可能交給其他角色接手，有待研議。

◎轉發者：由於設備那邊指定的 Trap/Syslog/Netflow 目的位址不方便時常變更，因此轉發者負責將來自設備的上述資訊根據來源設備 IP 即時轉發給對應的 ASD 接收，使 ASD 可以自由遷移位置依然能持續收到資料，所以 ASD 在遷移後必須通知轉發者新位置。

◎備份者：某些成員需要重啟或遷移時，如果需要快速回復狀態，可以先儲存回復資訊到備份者，然後取回以加速重構。

◎指揮者：主要任務是將來自管理者或使用

者的操作轉化成群組內操作，並回覆結果，目前暫時沒有賦予自動指揮任務的規劃。

以上是重要成員角色的大略說明，接下來說明這個架構如何解決我們目前面臨的問題。

1.3 解決的問題與結論

首先，ASD 的設計是針對設備獨立運行的，ASD 自己週期性對設備獲取資訊，同時也即時接收來自設備的 Trap/Syslog/Netflow，有狀況時可以快速反應且立即去設備取回相應資訊，各監控項目可以自行計算動態臨界值，自己作簡易異常判斷，自己評估惡化趨勢，許多工作都不需要頻繁動用核心資料庫，對核心資料庫或對設備都不會造成效能瓶頸。在監控項目一切安好時甚至不會打擾到其他角色，由於本身是物件導向程式打造的物件，很容易轉換成 ES 所需的 JSON 文件，抄錄自身資訊儲存備份或重啟回復、以及週期性記錄網管資訊到 ES 的歷史資料都很容易，一舉解決我們現行面臨的很多問題。其中最需要運算資源的動態臨界值計算，由於 ASD 可以自由遷移於計算雲內，不需要全部困在同一地同一台機器中，而且可以利用 ES 計算歷史資料，ASD 模組僅需作前置與後續處理即可，將運算力需求轉嫁給更加擅長大資料演算的 ES，而本架構設計也適合執行於運算雲中。

ASD 對監控項目的架構是樹狀物件，對每個監控項目與數值都能產生路徑做為定位值，而 RMD 關係管理者在建構關係網資訊的時候可以使用這些路徑值來記錄描述關係成員，當不同 ASD 的監控項目在關係網中被關聯在一起後，在異常綜合判斷時，很容易就能順著被牽手的樹枝爬到其他 ASD 並往上爬，對於異常根源的智慧化尋找、障礙事件影響服務範圍、嚴重性等級自動判斷都很有幫助，比從原始的大資料中去學習關聯快多了，甚至可以協助網管系統即時評估智慧化自動反應處理機制是否該執行，達到快速反應、縮短障礙時間的目標。

1.4 未來展望與規劃

依照 ASD 模組分散自治架構的特性，將規劃在各個模組導入 AI 邊緣運算，邊緣運算位於最接近監控資料來源的小型計算中心，主要功能在於收集、儲存、過濾、擷取、簡單的運算，並將處理過的資料與雲端系統進行有效率的交換，使系統變得更加即時、彈性且具有效率，依據監控標的所在位置與網路服務特性，加入 AIOps 機器學習機制(Machine Learning;簡稱 ML)可以動態調整監控標的告警值[3]，在 ASD 模組內可更彈性智慧化監控所屬監控標的物；在指揮者角色導入 AIOps 維運機制，搭配 EMD 與 RMD 的資訊，透過 AIOps ML 機制學習障礙告警事件，彙整關聯告警資訊，提供根因分析(Root Cause Analysis；簡稱 RCA)，當障礙發生時，收到眾多 ASD 告警資訊與 EMD 障礙事件，可快速彙整障礙資訊，提供障礙原因，快速解決障礙事件，將分散自治型網管系統導入 AIOps 示意圖如**圖2**所示，利用邊緣運算與機器學習特性，提升為智慧化分散自治行網管系統，未來可加入人工智慧機制，達成自動化修復與維運的功能。

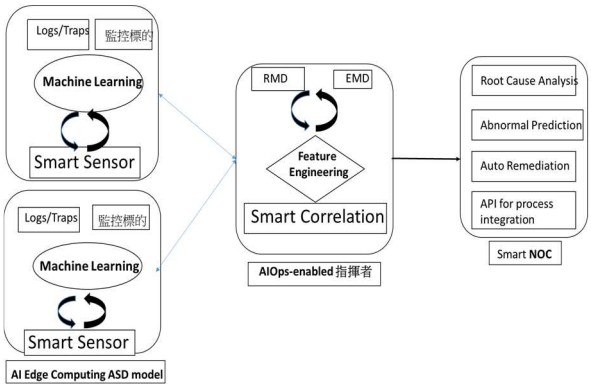

圖 2 智慧化分散自治型網管系統示意圖

參考文獻

[1] TaiWan Advanced Research and Education Network, http://www.twaren.net/.

[2] 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” in TANet2009, 2009.

[3] Masood A., Hashmi A. (2019) AIOps: Predictive Analytics & Machine Learning in Operations. In: Cognitive Computing Recipes. Apress, Berkeley, CA