大量家庭收視戶資料串流即時搜集與處理服務系統
A Realtime TV Audience Behavior Context Stream Analyzing System

吳伯彥、廖峻鋒
國立政治大學資訊科學系
Po-Yen Wu, Chun-Feng Liao
Department of Computer Science
National Chengchi University
Email: 102703045@nccu.edu.tw, cfliao@nccu.edu.tw

摘　要

隨著科技進步，為因應龐大資料的運算，雲端與大數據運算技術近年來蓬勃發展。另一方面，智慧家庭的應用也因為硬體技術成熟而受到重視。智慧家庭許多應用和雲端與大數據運算非常相關，例如整合雲端和家庭端，持續不斷地將使用者收視行為訊號化並傳回雲端分析中介軟體就能夠對用戶的收視行為進行自動化即時資料搜集與分析。然而，此種系統目前在架構設計上較為複雜，涉及多項異質如 MQTT 等通訊協定及 Apache Spark 與 Hadoop 等雲端中介軟體的互動與整合，且目前較少有這方面的研究。因此，本系統擬針對此一問題，本展示設計了一個整合從家庭網路到雲端網路，希望能提供有線電視服務營運商更多即時收視資料分析功能，讓營運商可進行更準確的市場評估，進而提供更貼近使用者的服務。

一、緒論

隨著近年軟硬體技術快速發展，雲端與大數據運算越來越被重視，大家紛紛追求更快、更穩定的架構。除了資料處理外，即時呈現也成為大家努力一個目標。智慧家庭許多應用和雲端與大數據運算非常相關，其中一個重要的應用就是針對用戶的收視行為進行自動化即時資料搜集與分析。為達成大量資料的即時事件處理，需要採用串流式的連續操作模型(continuous operateor model)，連續操作模型在容錯處理上通常採用備份的方式，當有節點出現錯誤，它會將資料重新操作一次，這種方法會使硬體成本加倍，且在恢復的過程，會使的整個系統必須等待，直到新的節點重新操作完成。因此我們採用 Spark Streaming 離散型串流(Discretized Stream)，本計畫期望能以 Spark Streaming 為即時處理串流，結合智慧家庭，建立一個即時收視戶資料處理架構，也希望藉此計畫能夠使得智慧家庭在未來能有更多的應用。

二、相關研究

目前在串流處理的技術上已有很多開發框架，而 Spark Streaming 是近幾年才新開發的框架。Apache Spark 是一個開源的叢集運算框架，Spark 使用了記憶體內運算技術，能在資料尚未寫入硬碟時即在記憶體內分析運算。而 Spark Streaming 則充分利用 Spark 核心的快速排程能力來執行串流分析。近年國內外有許多對於 Spark Streaming 的研究，像是基本的 Zaharia 等人(2012)研究 Spark Streaming 內容並介紹其內部運作與實際的效能表現的技術簡介。另外 Hunter 等人(2013)將 Spark Streaming 應用在交通運輸量上，實作出一個能在近乎即時更新的大規模的估計系統，讓使用者朝詢時能隨著現實交通狀況的不同，得知準確的行程時間。本研究將嘗試整合各段不同協定，使得資料能即時進入串流端，讓 Spark 核心能即時處理及呈現資料。

而 MQTT(Message Queuing Telemetry Transport)是 IBM 和 Eurotech 共同製定出來的協定，它透過 Publish/Subscribe 的訊息傳送模式，來提供一對多的訊息分配。Tang 等人(2013)利用 MQTT 協定，實作出推播功能，使得行動端的用戶只要程會 Subcriber，就會收到任何 Publisher 想要傳送的訊息， 另外在 Ghobakhlou 等人(2013)的研究中，網路系統與無線感測系統也透過 MQTT 協定來溝通，本次計畫參考以上研究，也希望能運用 MQTT 技術建立 Spark Streaming 與 CWMP 之間的傳遞溝通。藉此，我們希望應用上述技術 Spark Streaming 與 MQTT，結合在本研究上。本研究將嘗試整合各段不同協定，使得資料能即時進入串流端，讓 Spark 核心能即時處理及呈現資料。

三、系統設計與實作

3.1 系統設計

- 用戶端：每當用戶產生新的收視資料，則藉由智慧裝置自動將資料藉由 MQTT 協定，Publish 到 Server 端的 Broker。
- MQTT Broker：利用 Mosquitto 套件實作 Broker，一旦接收到來自用戶的 publish，便將資料廣播給 Spark Streaming。

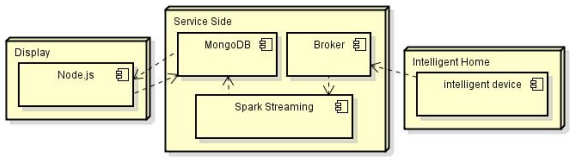

圖 1 架構示意圖

- Spark Streaming：藉由 streaming 的 MQTTUtils 中，提供的 createstream method 創建 MQTT stream，並將 Broker 的 URL 及 Topic 作為參數傳入，即可成功訂閱該 Topic；而在 ACS 端，我們使用 PAHO 來架設 client，並傳入 Broker 的 URL 與 Topic，即訂閱該 Topic。
- Display 呈現：藉由 Node Express 建置一個 Server，並利用 D3.js 將 MongoDB 中的資料視覺化呈現。

收視計算的部分，我們假設每個用戶每次操作智慧電視皆會產生一筆資料，資料總共包含四個欄位 customer_id，timestamp，period，channel_id 其中 timestamp 的單位是秒，是指與 1970/1/1 的時間差若正在收看 period 則為 0，若為關機狀態，channel 為 0)。首先創建一個 MQTT Stream 接收資料，一旦收到資料，首先判斷該筆新資料的 id 是否已存在，若不存在則直接將該筆資料新增至 RDD 中，若存在，則先將所有 id 與它相同的資料過濾出，並依照 timestamp 由大至小排序，找出最後一筆 period 為 0 的資料，由時間相減計算 period，並更新該筆資料，最後再把新的資料新增至 RDD 中，這樣的做法能讓每個 id 的資料至多只會有一筆 period 為 0 的資料，意謂使用者目前現在的狀態。在收視率計算時，我們首先定義收視率的分母為在該時間內收看電視的用戶數，分子為收看某頻道的客戶，而收視超過五分鐘才算入收看客戶，我們藉由收視資料 RDD 透過 filter 操作，篩選出所有在該時段收看的客戶數，以及收看特定頻道的客戶數，兩者相除即可得知該時間該頻道的收視率。

3.2 系統實作

客戶端的部分，我們利用程式隨機產生資料，並透過 PAHO Client 藉由 MQTT 將資料傳送到 Broker，藉以模擬實際收視戶。收視計算的部分如圖 2，首先定義一個 class 名為 Customer，其包含 customer_id, timestamp, period, channel 四個值，且將先前資料存入 dataRDD 中。之後每次要插入一筆資料時，將客戶端的資料轉為 Customer 的 object(period 預設為 0，簡稱 newest_data)， 存入暫時的 RDD，而在要存入資料庫前，會進行 period 計算，首先將資料 groupby customer_id，並由

圖 2 單一頻道時段收視統計

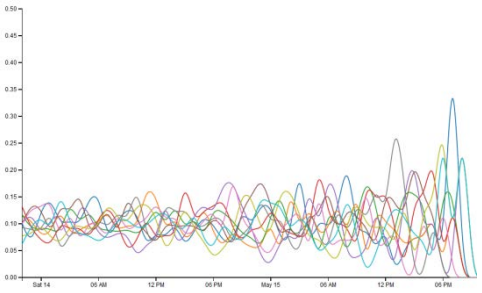

圖 3 頻道兩日收視率變化統計

timestamp 排序，再執行對每個 ID 的資料群進行 map，timestamp 依序相減，便可得到 period，再將資料存入資料庫中。Node Express 則由 mongoose API 與 MongoDB 進行連接，根據客戶需求撈資料，並透過 D3.js 畫出個別頻道的時段統計(圖 2)及所有頻道的折線圖(圖 3)。

四、結論

本展示期望能以 Spark Streaming 即時串流處理為核心，與智慧家庭結合，打造一個即時收視戶資料處理系統並藉由更直覺與即時的呈現，讓使用者準確地接收到資訊。也希望我們建立出來的架構能使的服務供應商能更即時的掌握收視狀況，進而提升服務品質，讓服務更貼近使用者。

參考文獻

Ghobakhlou, A., Kmoch, A., & Sallis, P. (2013, December). Integration of Wireless Sensor Network and Web Services. In *Proceedings of the 20th International Congress on Modelling and Simulation, Adelaide, Australia* (Vol. 16).

Hunter, T., Das, T., Zaharia, M., Abbeel, P., & Bayen, A. M. (2013). Large-scale estimation in cyberphysical systems using streaming data: a case study with arterial traffic estimation. *Automation Science and Engineering, IEEE Transactions on*, *10*(4), 884-898.

Tang, K., Wang, Y., Liu, H., Sheng, Y., Wang, X., & Wei, Z. (2013, October). Design and implementation of push notification system based on the MQTT protocol. In *2013 International Conference on Information Science and Computer Applications (ISCA 2013)*. Atlantis Press.

Zaharia, M., Das, T., Li, H., Shenker, S., & Stoica, I. (2012, June). Discretized streams: an efficient and fault-tolerant model for stream processing on large clusters. In *Proceedings of the 4th USENIX conference on Hot Topics in Cloud Ccomputing* (pp. 10-10). USENIX Association.