

資料串流上連續型查詢處理技術之研究(3/3)
Research on Continuous Query Processing Techniques over Data streams

計畫類別： 個別型計畫 整合型計畫
計畫編號：NSC-96-2221-E-004-002-
執行期間：94年08月01日至97年07月31日

計畫主持人：陳良弼

成果報告類型(依經費核定清單規定繳交)： 精簡報告 完整報告

本成果報告包括以下應繳交之附件：

- 赴國外出差或研習心得報告一份
- 赴大陸地區出差或研習心得報告一份
- 出席國際學術會議心得報告及發表之論文各一份
- 國際合作研究計畫國外研究報告書一份

處理方式：除產學合作研究計畫、提升產業技術及人才培育研究計畫、列管計畫及下列情形者外，得立即公開查詢

涉及專利或其他智慧財產權， 一年 二年後可公開查詢

執行單位：國立政治大學資訊科學系

中華民國 97 年 10 月 31 日

中文摘要

由於資料串流的特性迥異於傳統資料，再加上許多新興應用的推波助瀾，使得資料串流成為備受矚目的資料型態，舉凡資料串流管理系統（DSMS）的相關理論或應用雛型，皆為當前資料工程領域中熱烈討論的研究課題。為了滿足各種不同應用的實際需求，並且探討與解決現存資料串流管理系統所遭遇的種種問題與技術瓶頸，本計畫以研究 DSMS 的核心技術—連續型查詢（continuous query, CQ）之處理為主軸，開發此核心技術所需的各種關鍵技術；同時，本計畫更進一步地實際考慮感測器網路（sensor network）的環境，透過感測器特性之分析，發展感測器網路下之多查詢處理技術，並建立一實作系統作為使用者監測之用。

Abstract

During the last couple of years, many researchers have paid their attention to data stream management, which is much different from the conventional database management. At present, the new type of data management system, called data stream management system (DSMS), has become one of the most popular research areas in data engineering field. To satisfy all kinds of requirement in DSMS, we focus on the core technique – continuous query processing and have developed the key techniques for the concerned applications. Moreover, we develop the multiple query processing techniques for the sensor network by considering the real environment and the characteristics of sensors. Furthermore, a robust system is constructed for users to efficiently monitor the sensor network.

目錄

一、前言	4
(一) 資料串流之分析與監控	4
(二) 串流資料之連續型查詢處理	4
(三) 感測器網路之查詢處理	5
二、研究目的	6
(一) 資料串流之分析與監控	6
(二) 串流資料之連續型查詢處理	6
(三) 感測器網路之查詢處理	7
三、文獻探討	8
(一) 資料串流之分析與監控	8
(二) 串流資料之連續型查詢處理	9
(三) 感測器網路之查詢處理	10
四、研究方法暨成果展示	12
(一) 資料串流之分析與監控	12
(二) 串流資料之連續型查詢處理	16
(三) 感測器網路之查詢處理	23
五、成果自評	30
參考文獻	32

目錄圖表

圖一：新式概要結構.....	12
圖二：記憶體空間及準確率之實驗結果.....	12
圖三：傳輸策略示意.....	13
圖四：傳輸策略間的傳輸次數比較.....	14
圖五：傳輸策略間的探勘時間比較.....	14
圖六：有項目標幟的有序樹T及其對應編碼.....	15
圖七：與StreamT效能比較.....	16
圖八：每t個時間點的CM sketches.....	16
圖九：CM sketches記錄指數統計圖資訊.....	16
圖十：事件序列篩選系統之流程圖.....	17
圖十一：在不同查詢數量下我們方法所需之處理時間.....	18
圖十二：在不同查詢數量下暴力法所需之處理時間.....	18
圖十三：不同查詢數量下的總處理時間.....	19
圖十四：一至三條事件串流所需之總處理時間.....	19
圖十五：階層型聚合式網格索引架構.....	20
圖十六：不同移動物件數量下三種方法的效能.....	21
圖十七：不同查詢數量下三種方法的效能.....	21
圖十八：分群索引架構.....	22
圖十九：改變每筆資料所包含的平均項目數時的效能變化.....	23
圖二十：改變每筆資料的可能最大式樣時的效能變化.....	23
圖二十一：無線感測器感測排程技術之系統架構.....	24
圖二十二：能源節省比率及資料錯誤率.....	25
圖二十三：錯誤誤差與通訊錯誤率之關係圖.....	26
圖二十四：動態計數速寫資料結構.....	27
圖二十五：準確率實驗比較.....	28
圖二十六：所需空間實驗比較.....	28
圖二十七：MAKE DB 系統架構圖.....	29

一、前言

傳統的資料庫管理系統 (DBMS) 建構於可恆久存在的資料之上，對於這些資料，DBMS 可以提供穩定的儲存與索引架構，其查詢處理也允許多次掃描其資料庫。然而，在許多新興的應用中，資料往往會持續而快速地以串流的形式存在，這樣的特性導致傳統的資料庫管理技術難以適用，並使得與資料串流相關的課題逐漸成為資料工程領域中的重要研究趨勢。本計畫為三年期之計畫，著眼於各類應用對於串流資料處理與分析的急切需求，發展在串流環境下連續性查詢處理所需之核心技術，並且進一步考慮感測器網路環境，發展感測器網路下之多查詢處理技術，並建立一實作系統作為使用者監測之用。本計畫之研究成果可以歸納為下列三類：

(一) 資料串流之分析與監控

分析與監控資料串流的變動資訊可供查詢最佳化之用，也可作為研究、監測或者是決策之用，由於資料串流具有不定期更新和易快速累積之特性，以往將所有資料彙整存入資料庫系統，再定期以資料分析工具處理之作法已不可行。因此，現階段已經有不少研究投入資料串流的及時分析，其中以資料探勘與資料統計兩個領域的研究成果最為顯著：對於資料探勘領域而言，在串流型態的資料中作資料探勘，就如同以往在傳統資料庫上所從事的資料探勘一樣，也可發現許多有用的知識可供利用，但基於資料串流的特性，舊有的資料探勘方法都必須重新設計，才能克服在時間與空間上的嚴格限制；更特別的是，為了滿足實際應用的需求，也需考慮犧牲分析結果的精確性來換取時間與空間上的效益。而在資料串流的統計分析上，由於無法保留所有串流上流過的資訊，必須要引入滑動視窗的概念，在時間與空間的限制之下，要提供一個具有保證的統計值，使其與實際值的誤差在使用者所容許的範圍之內，才能夠真正滿足串流應用的實際需求。本計畫在資料串流之分析與監控部分的研究成果，一共包括了串流環境下之高頻樣型探勘 (frequent itemset mining) 技術、資料串流之樣型探勘 (pattern mining) 技術以及資料串流之統計值概算技術等，而這些技術可以用來支援目前已經開發出的各種表格資料之連續型查詢處理技術。

(二) 串流資料之連續型查詢處理

過去的研究對於靜態或是漸進式的資料庫已經提出很多不同類型的索引方法，藉以加速查詢比對的速度，並且配合各自的資料表示法、查詢表示法以及查詢處理演算法來找出滿足要求的答案，除了找出精確答案之外，有些方法更結合了相似度量測 (similarity measure) 來求出近似答案。然而這些方法面對的都是單次查詢 (one-time query)，只要先為資料庫中的資料建立索引架構再從中找出答案即可。由於面對串流資料時，資料是不斷地流入，若套用舊有的索引方法將會導致系統資源難以負荷，不但需要耗費大量時間進行索引架構的維護，也無法符合連續型查詢對於答案時效性的需求。針對此一課題，本計畫已發展出數個針對不同資料類型所形成之串流的查詢處理技術，以滿足不同的應用需求，這些技術包括了事件串流環境下之近似比對技術、移動物件環境下之 KNN 查詢方法以及交易串流環境下之區間查詢 (range query) 技術，這些技術可以有效地提升查詢處理的效率，讓串流環境下的各類應用服務品質得以進一步地強化。

(三) 感測器網路之查詢處理

除了一般性的資料串流環境外，由於感測器網路 (sensor network) 乃是一資料串流環境中之重大應用，為了與真正實際應用結合，我們將多查詢處理問題延伸至建構在感測器網路上之資料串流研究；也因為我們的研究真正的涉及實際應用的範疇，使得在實際情形下，感測器網路上的查詢處理技術，會受限於感測器本身之物理限制，如電力 (power)、傳送半徑 (transmission radius) 等，因而加深了資料管理之複雜度，卻也提供了不同的研究議題。針對感測器資料之近似聚合查詢問題，基於不同的設計概念，本計畫發展了兩個獨立的方法並且加以比較，而這兩個方法在效能上都勝過目前所存在的其他方法；同時為了應付實際的電力限制，本計畫分別設計了使用期限下之感測排程技術以及以趨勢分析為基礎之感測省電技術，並將這些技術一同彙整於實作系統之內；最後，為了減少多個查詢在聚合運算時的耗電量，我們也發展了一套計算分享的機制來進行多個查詢的最佳化處理。

茲就依照這三類主題，分別描述其研究目的、相關文獻、研究方法暨成果展示於以下各節，最後一節則為本計畫執行成效的整體評估與論文成果列表。

二、研究目的

(一) 資料串流之分析與監控

在資料串流的及時分析領域中，一個最重要的研究方向即為樣型探勘 (pattern mining)，其目的在於找出已經流過資料中出現比例高於某一門檻值 (threshold) 的樣型。在本計畫中，針對資料串流上資料探勘的領域，我們首先探討高頻樣型探勘 (frequent itemset mining) 的議題：現存的方法將原始資料串流建立成為一個概要 (synopsis)，並假設此概要能儲存於系統中，而忽略了非高頻樣型 (non-frequent itemset) 所佔用的空間可能造成系統資源大幅度地降低，因此我們在此研究中，結合了 Lossy Counting [MM02] 和 hCount [JQS03] 的原理，提出了一個新式的概要結構，使得原始的資料串流可以儲存在此一固定空間的概要結構中；基於此新式概要結構，我們再將問題延伸至多重交易串流的環境，欲解決多重交易串流之總體高頻樣型探勘 (global frequent itemset mining) 的問題，最直覺的方法，便是將分散式多重串流的資料，集結於單一電腦，模擬為單一串流，再套用單一串流之高頻樣型探勘演算法。然而，將多重串流的資料集結於單一電腦的動作，易使得單一電腦無法即時處理新進大量資料；因此，我們設計出一分散式系統架構，利用分散式探勘方式來探勘總體高頻樣型。

不僅止於分析由資料項目或是資料集合等簡單型態所組成之資料串流，我們也考慮到結構較為複雜的資料型態如 XML 文件所形成之串流，在本計畫中，我們針對樹狀資料探勘 (tree data mining) 領域中常見的資料型態—有標幟項目的有序樹 (labeled and ordered tree)，建立一可以快速增減子樹與更新其出現次數之資料架構與其存取技術，進而發展出一套可有效找出資料串流中所有高頻子樹 (frequent subtree) 的方法。

最後，針對資料串流的資料統計值計算，有鑑於資料串流具有流速快及單位時間內流量大之特性，我們需要排除過於緩慢的硬碟資料存取速度，以及必須要能夠在單次掃描的限制之下，盡快完成資料的更新與統計值的概算。在記憶體有限的環境之下，我們所發展的方法必須考慮犧牲結果的準確性來換取時間和空間的效益，然而在此考量之下，我們的方法依然可以保證其準確性的犧牲會落在使用者的容許範圍之內，以滿足使用者的確實要求。

(二) 串流資料之連續型查詢處理

隨著資料串流相關之應用快速增加，許多資料串流處理系統與相關技術便孕育而生，這些技術希望能在資料串流的環境中，得以有效率地處理大筆流入的資料，同時也能支援連續性查詢的需求。為了能夠處理更複雜的資料串流與查詢類型，以支援其他重要的應用，有別於以往只考慮表格資料之查詢處理，在本計畫中，我們亦考慮比表格資料更複雜的序列資料：即事件串流 (event stream)。在事件串流中，每一個事件可以包含不同個數的值，因此事件串流可以用來處理更為複雜的多媒體資料或是網路封包等；同時，在事件串流下，使用者通常感興趣的是某一段連續的事件是否發生，因此我們定義了一種新式查詢類型稱之為序列查詢 (sequence query)，讓使用者能夠持續偵測某一段連續事件是否出現。在現實情況下，有時使用者並不能下達完全精確的查詢，或者是資料串流在傳輸的過程中會有資料誤差的情況發生，因此我們必須要提供每一個查詢能夠獲得近似答案而非只有精確答案，如此一來才能夠符合實際應用的需求。為達成上述目的，我們開發了一能在事件串流環境中同時處理多個查詢的篩選機制，讓使用者能夠及時地得到滿足每個查詢容錯範圍的近似答案。

有鑑於移動物件環境也屬於串流資料環境中的一環，我們同時亦考慮在此環境中開發連續型 KNN 查詢處理技術。有別於過去的研究成果，我們將物件的屬性分類為位置相依型 (location-dependent) 與位置獨立型 (location-independent) 兩種，使用者能夠透過決定這兩種屬性的重要性來設計其個人化的花費測量機制，以取得滿足其需求之答案，因此我們所開發的移動物件環境下 KNN 查詢處理方法能夠滿足更為複雜的應用需求。

最後，像是市場販賣、使用者側寫檔以及網路使用資料等交易型資料 (transactional data)，也廣泛出現在許多新興的應用當中，這些資料通常都會大量且快速地累積，如何能夠及時地對這些資料做查詢處理也是一門重要的課題，本計畫針對此一課題，設計了嶄新的篩選機制以及資料結構，並據此開發一交易串流環境下之區間查詢技術，以符合這類應用所需的及時性要求。

(三) 感測器網路之查詢處理

在感測器系統應用中，除了單一感測器節點資料的收集外，針對一群感測器節點資料之聚合查詢 (Sensor Data Aggregation)，如平均、總和、記數、最大值等聚合值，也廣為使用，常見的感測器資料聚合查詢應用包括了平均雨量的回報、活動中之感測器節點數的回報等。現階段感測器系統中最常見的資料聚合方式為樹狀式資料聚合計算 (Tree-Based Aggregation) 與多路徑式資料傳遞 (Multi-Path Routing) 兩種，然而樹狀式資料聚合與多路徑式資料聚合各有其優缺點。在沒有網路錯誤發生的狀況下，樹狀式資料聚合擁有精確的查詢結果，多路徑式資料遞送則僅能提供近似或錯誤的結果。而在網路狀況不佳的條件下，樹狀式資料聚合將會遺失大部分的感測器觀測值，而多路徑式資料聚合則保有較佳的容錯能力，並提供一個較準確的聚合查詢結果。有鑒於此，本計畫以多路徑式資料遞送方式為基礎，設計出具高可靠度、高準確率與高能源效率之感測器資料聚合計算方法，以供感測器資料聚合查詢應用使用。

此外，在無線感測器網路系統上 (Wireless Sensor Network) 由於感測器之電力供應是由電池所提供，再加上無線傳輸和感測功能皆須消耗相當大量之電力，一旦電量耗盡，即失去功能無法再行充電，因此在增加電源供應的狀況下，如何節省電量消耗，是無線感測器網路研究範疇中，相當重要的課題之一。對此一課題，本計畫從兩個面向來加以探討：第一，在感測器網路續航力方面的研究，只著重於如何節省能源消耗量，而無有效方法能確切規劃感測器之使用期限，為了能夠達到使用者所預定的監測時限，方便監測計畫的訂定和付出成本的計算，我們藉由發展感測時程技術 (Acquisition Rate) 及突發事件應對之排程技術，來規劃具有『以達到任務期限為目標』之無線感測器網路，盡可能在不同電力供應下，滿足使用者查詢所需之特定突發狀況的同時，還能夠將剩餘的電量做有效之分配利用；第二，TinyDB 與 TAG 等感測器資料管理系統，其收集資料的方式為定期地偵測環境變數，並且回傳給資料管理系統，然而其偵測週期乃由管理者直接下達指定。若偵測時間設定較長，則較節省能源，但相對得到較少的環境資訊；若偵測時間設定過短，則相對應得到較多資訊，但卻十分耗電。在一般的感測應用中，如溫度監測等，常具有其週期性，過份密集的偵測不僅耗電也無助於取得有用資訊，基於此觀察，我們設計一週期探勘的演算法，針對感測器所偵測之歷史資料加以分析，探勘這些資料的週期資訊，並利用所找出的週期特性，以便動態調整偵測時間，達到省電的目的。

三、文獻探討

(一) 資料串流之分析與監控

高頻項目(Frequent Items)或高頻樣型探勘在資料串流上的相關研究主要可以依據使用者所決定的新近資料(Recent Data)重要程度，概略分為三種模型(Model) [LCW05]，分別是：滑動視窗模型(Sliding Window Model)，隨時間淡出模型(Time-fading Model)及地標模型(Landmark Model)。

在滑動視窗模型中的相關研究[CWY04][GDD03][LCW05][LK06]，強調的是位於滑動視窗中資料的重要性，當一筆交易移出滑動視窗之後，該交易對其相關樣型支持度的貢獻就會跟著移除。在隨時間淡出模型中的相關研究[CL03][GHP03][LL05]，強調的是時間敏感度，也就是越新進的資料的重要性，會較越早期的資料來的高。其中，在[CL03][LL05]內的衰敗機制(Decay Mechanism)就是一隨時間淡出模型的例子。在衰敗機制中，使用者可以根據新近資料的重要程度，來決定一衰敗比例(Decay Rate)。當一筆新交易產生的時候，新的串流資料量會等於舊的串流資料量乘上衰敗比例之後，再加上目前這筆最新交易；換言之，舊的串流資料量所佔的權重，會低於目前最新交易。除了衰敗機制外，傾斜時間視窗(Tilted-time Window)也是另一隨時間淡出模型的例子[GHP03]。在[GHP03]中，目前最新的交易會落在最小時間刻度(the Smallest Time Scale)的當前視窗中，而越過時的交易則會落在越大時間刻度的視窗中，這些視窗中的子高頻樣型(Sub-frequent Patterns)會利用近似於高頻樣型樹(FP-tree) [HP00]的資料結構儲存，以加速高頻樣型的探勘。

在地標模型中，會利用一地標時間點(Landmark)，來表示系統啟始時間，而使用者想獲得的高頻樣型，即是落在地標時間點到目前時間點之間的那些高頻樣型。在地標模型中的相關研究主要可分為虛假導向型(False Negative Oriented) [YCL04]及虛真導向型(False Positive Oriented) [CCF02][CM03][DLM02][JA05][JQS03][KPS03][MM02]之演算法。虛假導向型之演算法強調的是，由此類演算法所探勘而得的高頻樣型，一定是真正的高頻樣型，但並非全部真正的高頻樣型都包含在探勘結果中。另一方面，虛真導向型之演算法強調的是，此類演算法一定能探勘出所有真正的高頻樣型，但其結果可能同時包含非高頻樣型。[DLM02]和[KPS03]延伸了多數演算法(Majority Algorithm) [FS82]的精神，用其來發展高頻項目探勘的演算法。在這兩個研究中，皆利用一群特定數量的計數器(Counter)來記錄項目出現的個數。當一個新項目被產生時，若該項目已對應到某計數器，則該計數器的計數值便會增加；反之，若沒有任一個計數器能對應到該項目，且尚有其他計數器沒有對應到任何其他項目，則可將該項目對應至任一可用之計數器。倘若，所有的計數器皆有其對應之項目，而無存在任何可用之計數器時，則需將所有的計數器中的計數值都減少一，而計數值為零的計數器則不再對應到原本所對應的項目，重新變為可用之計數器。[JA05]中，Jin 等人則是將多數演算法的觀念用於探勘高頻樣型，發展了一個入核心(In-core)演算法。在此演算法中，長度為一的樣型，也就是項目，是全部儲存於記憶體中，而多數演算法是用來處理長度為二的樣型，至於長度為三以上之樣型，則利用 Apriori 特性[AS94]來產生。但在此方法中，必須利用一無限制大小的緩衝器(Buffer)來儲存串流中的交易，以便於產生長度較長之樣型時，能透過多次掃描這些交易來得到其支持度，較不適用於串流資料的線上探勘(Online-mining)。

在[CCF02]、[CM03]及[JQS03]中，則是利用雜湊基礎的方法於資料串流中探勘高頻項

目。在這些研究中，多半是讓項目經由幾個雜湊函數計算後，將其資訊存入雜湊表格中相對應的計數器中，使得一個項目會和多個計數器相互對應，而且多個項目也可能會對應到一共同計數器。在這些以雜湊為基礎的演算法中，雜湊表格的維護程序(Maintaining Procedure)非常簡易，當有新的項目被產生時，與該項目相關的計數器之計數值便會增加，而各個項目的支持度則可由它們各自相對應的計數器來估算。然而，以雜湊為基礎的方法到目前為止，僅提供於高頻項目的探勘。在[JQS03]內所提的 h 計數(hCount)方法中，當使用者想知道目前的高頻項目，也就是在所謂的探勘階段(Mining Stage)時，所有的項目皆會經過雜湊函數計算來求得其支持度，並一一檢查是否有滿足最小支持度。此外，Manku et al.在[MM02]中則是發展了一少算(Lossy Counting)演算法來探勘高頻樣型。此演算法主要的原理是，僅在系統中記錄較高頻的樣型，而且盡量快速地將低頻樣型給移除。然而，由於為了達到樣型之真正支持度和估計支持度之差值能小於使用者所定義的錯誤參數值(Error Parameter)；因此，該演算法必須將支持度超過錯誤參數的樣型全部儲存，而造成系統記憶體大幅被佔用。

在本計畫中，我們將分散式探勘系統建構成一連續分散式模型。所謂連續分散式模型[CMZ07]其架構如下：有多個遠端網點(Remote Site)，且每個遠端網點會監控一資料串流，而另有一伺服器(Server)其主要的功能是來作為所有遠端網點的協調者(Coordinator)，在此一分散式系統架構下，伺服器端會在所有資料串流的聯集上，持續監控一特定任務(Task)，我們可將之視為伺服器端會於所有資料串流的聯集上執行一特定計算函數(Computational Function)，如分群(Clustering) [CMZ07]、越門檻計數(Thresholded Counts) [KCR06]、複本回復(Duplication Resilient) [CMZ06]、前 k 監控(Top-k Monitoring) [BO03]、追蹤集合表示(Tracking Set Expressions) [DGG04]及分位數(Quantiles)監控[CGM05]等。在此模型下，多半不考慮由網路所造成的訊息延遲(Message Delay)、封包遺失(Packet Loss)或是同步(Synchronization)議題等，而假設在傳輸過程中，傳送者將資料送出的瞬間，接收者便可收到，進而將問題化簡成，監控結果正確性(Correctness)及準確性(Accuracy)完全取決與所設計之相關演算法。

(二) 串流資料之連續型查詢處理

包括 DNA 序列比對、內涵式複音音樂查詢以及資料串流管理等研究領域都與我們所開發的串流資料的連續型查詢處理有關，分別介紹如下：

在 DNA 序列比對的領域，大多是考慮由 A、T、C 與 G 四種單一符號所組成的基因序列，其中 BLAST[AG90][PL98]是最廣為人知的基因序列比對工具，這個工具雖然有極高的比對效率，但是確有可能會遺漏部分的答案，在其他應用環境之下，這種情況可能會導致服務品質的嚴重下滑，而我們研究的目標則是在進行連續型查詢處理時，不遺漏任何的答案；此外，[KS01]則是利用序列中所有的子字串利用小波轉換技術轉至數字空間，將所得之小波係數建立在索引結構之上以加速比對速度，然而這種方式並不適合用在串流環境之中，因為我們所考慮的多數值串流會比單數值串流更為複雜，這樣的轉換將會造成更大的搜尋空間，反而會拖累比對的效益。

針對複音音樂之查詢，因大部分的音樂查詢系統乃是以音高為比對的標準，而在複音音樂中同一個時間點會有一個以上的音符同時發生，因此一首複音音樂可以由一個多數值字串來表示。對於複音音樂之查詢，[LP00]根據shift-or演算法發展一新演算法，但是與shift-or演算法相同，其查詢樣型的長度受限於計算機中的字元長度，而且其查詢只限於單音的形式；

為了提供複音查詢的能力，[CE00]中提出以集合的方式儲存樂曲中音符出現的時間與該音符的音高，形成類似總譜表 (score-like) 的格式，依此來處理音樂查詢，但是這種技術只有在提供精準答案時具有效率；此外，[Dov99]則是提出一個以Dynamic Programming為基礎之演算法，希望能夠在音樂查詢時提供近似答案，然而由於該技術的限制，在音樂資料庫過大或是音樂字串過長時，會因為計算量過大而導致效率明顯降低。

在資料是獨自存在的串流管理的領域中，針對不同的應用環境與需求，已經有許多解決的方法被陸續提出[G003]，但是大多數的資料串流管理系統[CC02][CD00][HF00][MW03]考慮的則是 tuple-based 的資料串流，而且其查詢語言則是改良至 SQL 語法，在 tuple-based 的資料串流中，每筆資料之間並沒有直接的關連性，是各自獨立的[AH00][CF02][KN03][MS02]，當我們需要考慮到資料間的時間關係時，這些系統所採用的技術便無法滿足我們的需求。

(三) 感測器網路之查詢處理

在高可靠度無線感測資料聚合計算技術相關研究中，[MFH02]首先提出兩個策略，分別為子節點快取(Child Cache)與分數式傳遞(Fractional Parent)，來提升無線感測器資料聚合計算的準確度。在子節點快取策略中，每個節點皆儲存其子節點先前傳遞過的部份聚合值，當某個感測器節點之子節點資料遺失時，該感測器節點利用先前儲存之該子節部份聚合值進行資料聚合運算。然而，在感測器節點中，儲存子節點過去資料，將造成額外的記憶體花費與感測器能源消耗。在分數式傳遞，作者提出多路徑式資料傳遞的概念，廣播部分聚合值至其上一層的節點，並採用均分式切割(舉例來說，若某一節點有五個母節點則該節點將其部份聚合值除於五，並廣播該值至其所有母節點)，來減低多路徑式資料聚合所衍生的重複記數問題。然而，如同在[CLK04]中所指出，分數式傳遞的策略只減低了聚合查詢結果的變異數(Variance)，重複記數所造成的錯誤資料聚合結果並無因此改善。

在[CLK04]及[NG04]中作者提出利用複本無影響性之速寫結構(Duplicate-Insensitive Sketches)來解決多路徑式資料聚合計算中所產生的重複記數問題。其主要的精神在於利用在感測器傳遞資料時，利用複本無影響性之速寫結構來表示感測器所將廣播之資料。由於速寫結構之複本無影響性特性，相同的一筆資料最終將只會被計算一次，因此解決了重複記數問題。[NG04]正規地定義複本無影響性之速寫結構其數學上該有的性質，並提出使用複本無影響性之速寫結構來解決重複記數問題的標準架構。[CLK04]則提出使用 FM-Sketches [FM85]，來解決多路徑式資料聚合所產生的重複記數問題。利用[NG04]中複本無影響性之速寫結構定義，FM-Sketches 可被證明為複本無影響性之速寫結構。

給定一個多重集合，FM-Sketches 可快速地估算出此多重集合裡不同元素的數量。使用來 FM-Sketches 估算一個多重集合裡不同元素數量的操作方式如下：首先給定一條長度為 n 之初始值為零的二元陣列，接下來利用一組隨機雜湊函數，該組隨機雜湊函數模擬一連串二項式測試(Binomial Trail)，(連續地投擲一枚正反兩面公平之錢幣，連續投擲直到該錢幣的人頭出現)，將多重集合裡之元素逐一地對應到初始值為零的二元陣列，並將對應到的位址值設為壹。如此一來，多重集合裡不同元素數量 N ，則可利用二元陣列裡由左側數來最右側被對

應到的位元 R_N 來估算。FM-Sketches 估算的精神在於，若一個多重集合裡不同元素數量越多，則二元陣列裡最右側被對應到的位元 R_N ，也將會越大。因此利用二項式測試隨機雜湊函數的特性，多重集合裡不同元素數量可利用 $E[R_N] = \log_2(0.773 \times N)$ ，此公式來估算。

然而，使用 FM-Sketches 來解決多路徑式資料聚合計算所產生的重複記數問題，將會面臨三個主要缺點。首先，使用 FM-Sketches 來估算資料聚合值，所得的估計值準確率低且所得的資料聚合估算值的變異程度高。由於 FM-Sketches 所使用之雜湊函數模擬一連串二項式測試，在 FM-Sketches 二元陣列中由左側算來第 i 個位元被對應到的機率較第 $i+1$ 個位元被對應到的機率多出一倍。因此在估算中，二元陣列裡由左側數來最右側被對應到的位元 R_N 實際上各有一半的機率，會被對應於第 (R_N+1) 位元與第 (R_N-1) 位元，造成使用 FM-Sketches 準確率低且變異程度高的缺點。第二，FM-Sketches 的使用需在各感測器節點維持一組隨機雜湊函數，造成額外的記憶體花費與感測器能源消耗。第三，使用 FM-Sketches 無法在資料聚合估計值之準確率上有任何保證。

在[MN05]中，Manjhi et al.則提出在一個無線感測器網路中，同時使用多路徑式資料聚合與樹狀式資料聚合來進行高可靠度無線感測資料聚合；在網路通訊失敗率較高的地方使用以 FM-Sketches 為基礎之多路徑式資料聚合，而在網路通訊失敗率較低的地方使用一般的樹狀式資料聚合。這樣的方法巧妙地結合多路徑式資料聚合與樹狀式資料聚合的優點，然而，使用此方法的主要代價為需主動地管理並動態切換感測器網路中各區域所需維持之資料聚合模式。當網路狀況不穩定時，頻繁地切換各區域所維持的資料聚合模式，將造成額外的能源消耗與能源利用率的下降。

在[SB04]中，Shrivastava et al.則探討無線感測器網路中較複雜的資料聚合函式的計算，例如中位數(Median)與統計條狀圖(Histogram)。論文中假設感測器網路中並無網路通訊失敗的情形存在，並將焦點置於感測器觀測值的統計資料儲存結構之設計。在[CP06]中，Chen et al.則提出利用重複的隨機資料交換程序(Data Exchanging Process)的方法來計算無線感測器網路中資料聚合值。在一次隨機資料交換程序中，網路中的感測器節點隨機的成為主導節點。成為主導節點的感測器節點利用廣播邀請其鄰居來形成感測器群組，群組中的節點將其所儲存的部分資料聚合值傳遞給主導節點。主導節點接收其群組所傳遞之部分資料聚合值後，計算新的部分資料聚合值，並廣播予群組中之節點。利用如此的隨機資料交換程序，各感測器節點所儲存的部分資料聚合值將逐漸收斂至正確的資料聚合值。然而使用這樣方法的缺點為需要多次的隨機資料交換程序，造成感測器能源的消耗與查詢結果回報的延遲。

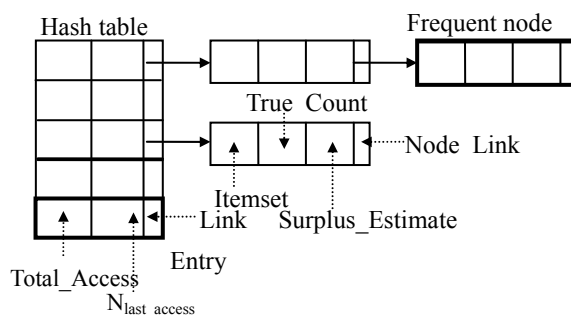
四、研究方法暨成果展示

(一) 查詢串流與資料串流之監控

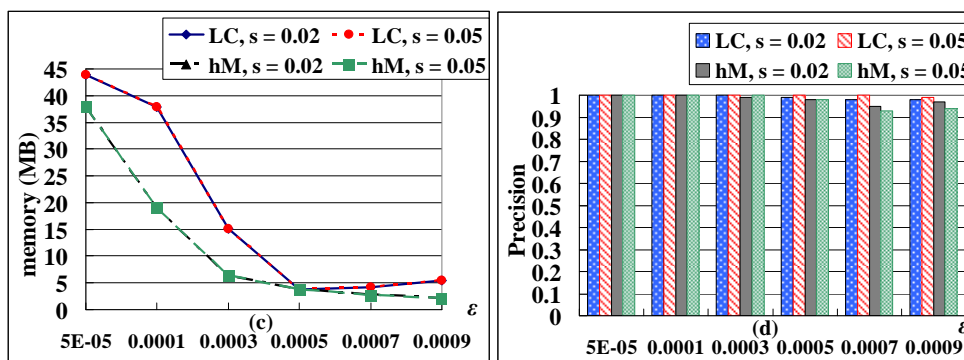
以下分別介紹在本項目中所開發的四項技術及其成果：

1. 高頻樣型探勘方法

在Lossy Counting的方法中，其概要必須將支持度 (support) 大於使用者定義之錯誤容許參數 (error parameter) 的所有樣型儲存下來，以達到樣型的估計支持度跟真正支持度的差異，能保證在小於錯誤容許參數之內。但因為錯誤容許參數遠小於最小支持度 (minimum support)，使得真正支持度落於錯誤容許參數和最小支持度之間的樣型數量，太過龐大，進而造成概要所需儲存空間過大。相對而言，在hCount的方法中，提出一略圖 (sketch)，來將原始資料串流的資訊壓縮於一個固定空間內，並在回答高頻項目 (frequent items) 時，將所有項目一一檢查。因此，若將hCount的方法直接延伸來處理高頻樣型探勘的問題，在利用該略圖檢查每個樣型的支持度是否大於最小支持度時，會耗費相當多的時間。因此針對上述兩個方法的缺點，我們設計一新式概要，如圖一所示，此概要由兩個部分組成：



圖一：新式概要結構



圖二：記憶體空間及準確率之實驗結果

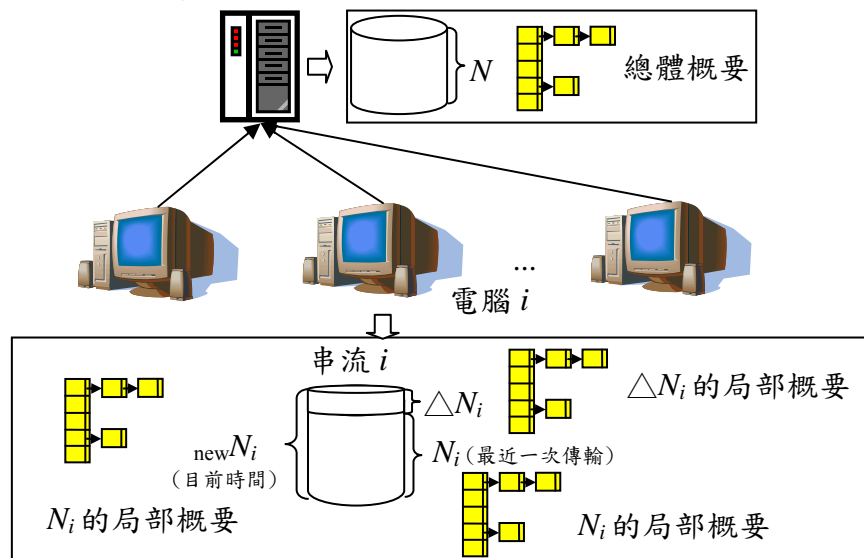
第一部份為雜湊表格 (hash table)，利用一雜湊函數 (hash function) 將原始串流資料中每個樣型的資訊，完整記錄在此雜湊表格中；第二部分為高頻節點 (frequent node)，將所有高頻樣型的資訊，另外儲存在高頻節點中，以期加速高頻樣型的探勘。就某一特定樣型而言，若該樣型儲存於高頻節點中，則我們可根據高頻節點上的資訊來決定該樣型的支持度；但若此樣型沒有紀錄於任何高頻節點中，我們可以根據雜湊表格中所記載之資訊，回估此特定樣

型的支持度。

實驗的成果如圖二所示，我們所提出的新式概要在記憶體的需求上，其表現都明顯優於 Lossy Counting。

2. 分散式多重串流下之高頻樣型探勘方法

如圖三所示之分散式系統建構，我們利用一台電腦來處理一局部串流 (local stream)，並另設一伺服器 (server) 來當作所有電腦的協調者 (coordinator)。對於一局部串流而言，處理該串流的電腦利用我們在上一項目中所設計之雜湊式概要結構，建立此一局部串流的概要。而對於伺服器而言，它主要的工作是在於合併 (merge) 所有局部串流概要，形成所有局部串流的聯集 (union) 所對應之一總體概要 (global synopsis)。此外，為求節省電腦端將其局部概要傳送至伺服器，且伺服器合併所有局部概要所花費之等待時間，我們將此一分散式系統建構成一連續分散式模型 (continuous distributed model)，使得伺服器所維護之總體概要皆永遠對應至目前所有局部串流的聯集。

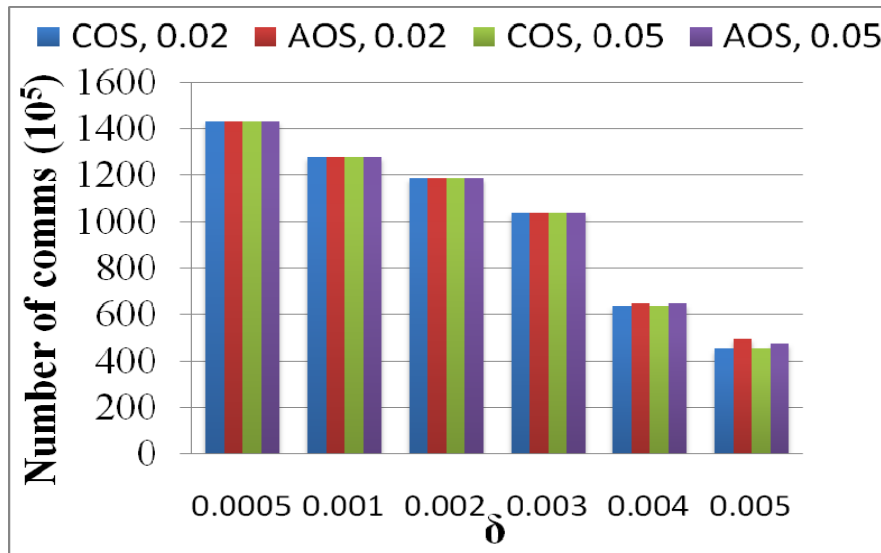


圖三：傳輸策略示意

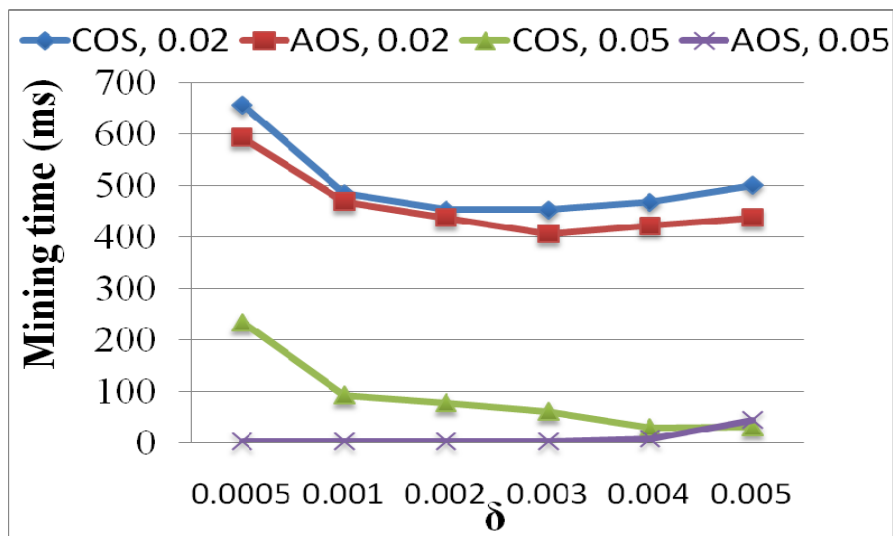
在此分散式系統架構中，我們採取單向傳輸策略 (one-way communication strategy)，也就是伺服器端僅接受由電腦端傳來之資料，而不會傳送資料至電腦端，藉此來節省由伺服器端傳送至電腦端所可能產生之傳輸量。在電腦端，我們設計兩傳輸策略 (communication strategy)：一、COS 策略：僅考慮根據使用者對樣型所給定之真正支持度與估計支持度的差異容忍範圍以及二、AOS 策略：除上述考量外，同時考慮樣型狀態的改變，例如：由高頻樣型轉變為非高頻樣型，或由非高頻樣型轉變為高頻樣型等因素。當超過上述傳輸策略裡訂立的傳輸限制時，強迫此一電腦傳送相關資料給伺服器。

在電腦端強迫與伺服器溝通時，電腦端傳送至伺服器的資料為：上次傳輸後至目前時間點的概要差異，也就是如圖一所示，我們需將 ΔN_i 的局部概要傳送至伺服器。根據傳輸策略的特性，我們設計兩個策略來產生 ΔN_i 的局部概要：一、直接建立新的概要來維護 ΔN_i 的局部概要及二、利用目前最新的局部概要及最近一次傳輸時的局部概要來推導 ΔN_i 的局部概要。此外，伺服器端的工作十分簡易，由於在此分散式系統上是使用單向傳輸策略，所以伺服器端的工作便是接收由電腦端所傳送之 ΔN_i 的局部概要，並將其結合至總體概要中，持續維持目前所有局部串流聯集所對應之總體概要即可。

圖四顯示兩傳輸策略COS和AOS之間準確度(Precision)的比較，很明顯的因為AOS傳輸策略設定的關係，使得AOS的準確度較COS來的略高，但如同圖三所示，AOS的傳輸次數略高於COS的傳輸次數。如圖五所示，由於AOS策略中，總體概要所記載的高頻樣型較COS策略來的精準，而探勘時間是取決於所記載之高頻樣型個數，因此，在探勘時間的表現上，並沒有AOS絕對優於或劣於COS的結果。



圖四：傳輸策略間的傳輸次數比較



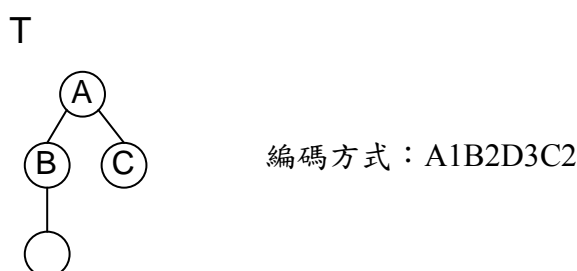
圖五：傳輸策略間的探勘時間比較

3. 頻繁子樹樣型探勘

我們所發展的頻繁子樹樣型探勘方法一共可分為四個步驟，分述如下：

- (1) 樹之前置處理與編碼：我們所探討的查詢串流是由多筆查詢樹組成，查詢樹是一種有項目標幟無序樹 (labeled unordered tree)，有項目標幟無序樹上的節點 (node) 具有下述特性：每個節點具有唯一的項目標幟 (unique label)，同一父節點 (parent node) 的所有節點沒有次序之分。為了避免同一棵無序樹產生大量的同構子樹 (isomorphic subtree)，增加儲存和計數上的複雜度以及系統負載度，我們先將每棵查詢樹置入一正規化程序，其過程如下：即從樹根開始，依廣度優先順序與字元次序對該節點的所有子節點排序，所得之有序樹即稱為正規

樹。同時為了便於產生所有子樹及計數等步驟，正規樹需再經過轉換成為特殊編碼的字串形式。此編碼方式僅能得到唯一的編碼字串，同時亦可將編碼字串轉換回原本的正規樹，此編碼過程為：依深度優先順序走訪正規樹中的各節點，依此順序列出其項目標幟以形成字串，如圖六所示。



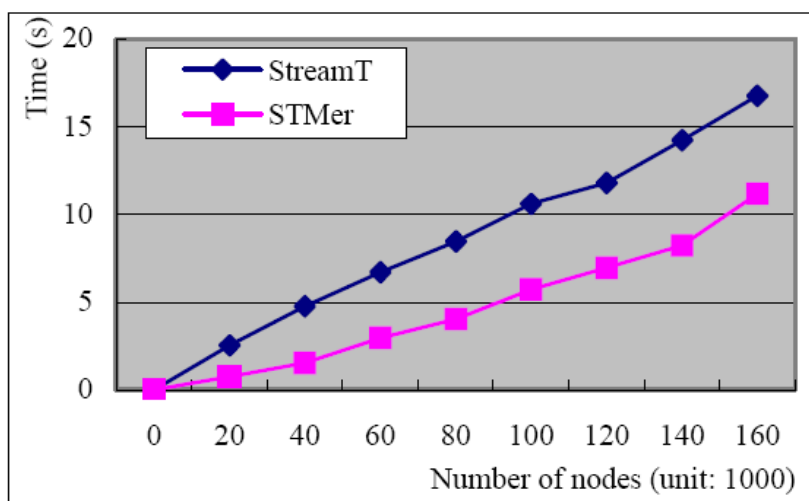
圖六：有項目標幟的有序樹 T 及其對應編碼

(2) 列舉所有子樹並合併至全域字首樹 (Global Prefix Tree)：在此步驟，我們可根據編碼字串的特性，制訂的增加字首樹 (APT, Augmented Prefix Tree)。其實因為在編碼的設計過程中，我們有將各節點所在的高度 (Level) 明確的紀錄下來，再加上編碼的過程是利用深度優先 (depth first) 順序，因此以這兩項特性反推我們可以建立一增加字首樹用以快速列舉並儲存所有的子樹樣型，之後再將其合併 (merge) 至全域字首樹中。

(3) 以 Lossy Counting[MM02]演算法為基底，刪除計數非常小的子樹樣型：若我們將每棵查詢樹的增加字首樹找出，合併至全域字首樹，在資料流經一段時間後，可能會有非常龐大的全域字首樹，造成記憶體空間不足；因此，我們將全域字首樹的建置值基於 Lossy Counting[MM02]演算法上，當固定幾棵查詢樹加入全域字首樹之後，就將計數不足其規定的節點加以刪除，以達到節省記憶體之原則，並維持少算的部分能在容忍誤差範圍內。

(4) 頻繁子樹探勘處理：當使用者要求回傳目前頻繁子樹時，我們對全域字首樹作深度優先的走訪，依據該子樹的計數值佔整體資料串流大小之比例是否超過使用者給定之最小支持度 (Minimal Support)，回傳頻繁且被標記為封閉的節點。

我們利用由Zaki所提供之資料產生器產生一組資料來進行實驗的測試，用來比較我們的方法 (STMer) 與過去的方法 (StreamT) 效能的優劣，結果如圖七所示，在資料量不斷增加的情況之下，我們的方法大致上都只需耗費原本方法 7 成左右的時間而已，所以可以顯著地提升在串流環境中頻繁子樹樣型探勘的效能



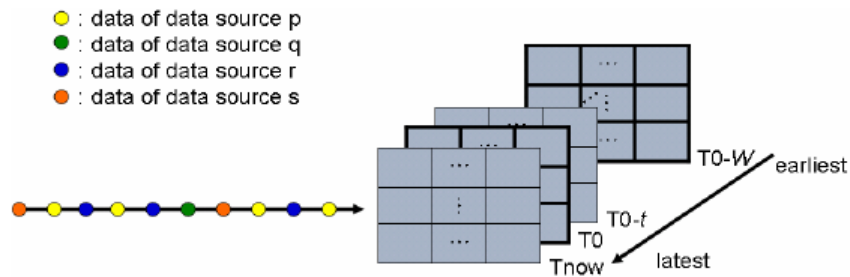
圖七：與 StreamT 效能比較

4. 多資料串流之統計值概算

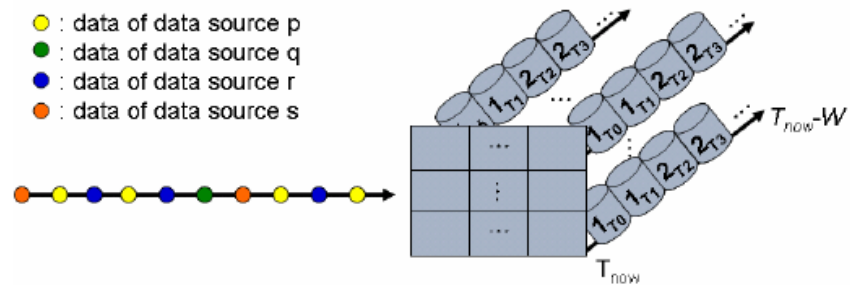
為了處理資料量龐大的串流資料，我們以 CountMin (CM) sketch [CM05] 的結構，來儲存大量累積資料之摘要 (summary)，以期只需依據 CM sketch 的內容便可概算滑動視窗內的所有統計值資訊，且能保證這些結果都可落在使用者所訂立的誤差範圍內。在此研究成果中，我們基於 CountMin sketch 的結構，以不同的觀點，發展了二種方法來計算移動總和，分述如下：

(1) 離散方法 (The discrete method)，在本方法中，每經過 t 個時間，我們就會計算此時間段中各資料來源流入的資料，因此，在長度為 W 的移動視窗中，共有 W/t 個不同的資料片段，如圖八所示。當視窗移動的時候，只需把最舊的 t 時間內的總和資料移除，再記錄新進的總和資料即可。

(2) 連續方法 (The continuous method)，在本方法中，使用指數統計圖 (exponential histogram) 的技術，記錄目前最近的移動視窗中，所有流入資料的時間戳記 (timestamp) 與其數值，如圖九所示。並藉由指數統計圖內資料的時間戳記，刪除已離開目前移動視窗的資料，來確保指數統計圖內的資料一定是最近 W 時間內的資料，再利用記錄之數值來計算移動總和。



圖八：每 t 個時間點的 CM sketches



圖九：CM sketches 記錄指數統計圖資訊

(二) 串列資料之連續型查詢處理

以下分別介紹在本項目中所開發的三項技術：

1. 事件串流環境下之近似比對技術

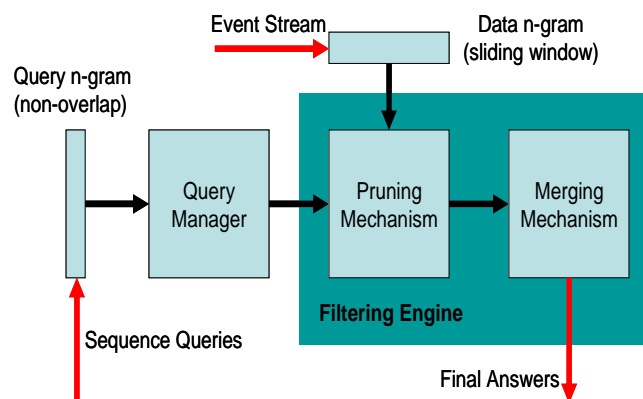
我們所發展的事件串流環境下之近似比對技術一共可分為三個步驟，分述如下：

(1) 將資料與查詢分別前處理為資料 n -gram 與查詢 n -gram：為了降低系統的運算量以符合查詢的及時需求與系統資源的限制，我們所發展的事件序列篩選系統採用之根本概念乃是利用

查詢間存在的共通性，讓相似的查詢能夠一起進行比對的工作，進而節省運算工作量。圖十為我們系統運作的流程。查詢管理者 (query manager) 的工作是對所有註冊的查詢建立索引架構，我們將每個查詢分割成數個不重疊且長度為n的片段，稱為查詢n-gram (query n-gram) 再套用一分群方法將相似的查詢 n-gram放在同一分群之內。同時，為了加速查詢比對的工作，每一個分群都會建立一個摘要 (summarization) 以代表該群，作為和每一筆流入資料比對之依據。另外，我們利用一流動視窗 (sliding window) 擷取長度為n的資料片段，稱為資料n-gram (data n-gram)。當查詢索引建立完成後，刪除機制 (pruning mechanism) 會開始針對事件串流所流入的每一筆資料 n-gram進行比對，以找出哪一筆資料 n-gram有可能成為某一查詢答案的一部份。

(2) 資料 n-gram 和查詢 n-gram 的比對：我們提出了一個計算資料 n-gram 與每一分群摘要間限制編輯距離 (restricted edit distance) 的方法，讓系統得以快速地驗證目前流入的資料 n-gram 一定不會是哪些分群的答案。如果一個資料 n-gram 與一個分群的最短距離大於每一個屬於該分群的查詢誤差容忍值的話，我們便可以將該分群的所有查詢 n-gram 忽略；反之，我們將該資料 n-gram 送至每一個相對應的查詢緩衝器 (query buffer)，以便於在組合答案的時候使用。

(3) 合併查詢答案：合併機制 (merging mechanism) 的工作在於嘗試組合出能夠滿足某一查詢的近似答案。原則上，對於一個查詢來說，只要我們能夠找到滿足其每一個查詢 n-gram 的資料 n-gram 時，我們就可以試著組合出一個可能的近似答案。然而如此鬆散的合併機制，將會浪費許多不必要的計算。因此，根據串流的特性以及誤差容許值的限制，我們在合成近似答案時採用了四項檢查，來避免組成一些不可能的答案，導致檢查時間的額外花費，這些檢查包括了滿足第 k 個查詢 n-gram 的資料 n-gram 必須在滿足第 k-1 個查詢 n-gram 的資料 n-gram 存在緩衝器中才屬有效；當組合出部分解答時，該部分答案的長度不能違反誤差容許值的限制；實際檢查目前資料 n-gram 與查詢 n-gram 的最小可能誤差，確定其沒有違反誤差容許值的限制；最後，在合成部分解答時，利用評估的方式預測已經產生的最小可能誤差，確定其沒有違反誤差容許值的限制。當我們完整組合出一段答案時，便要實際執行與查詢的距離計算，以確認是否為真正滿足之近似答案。

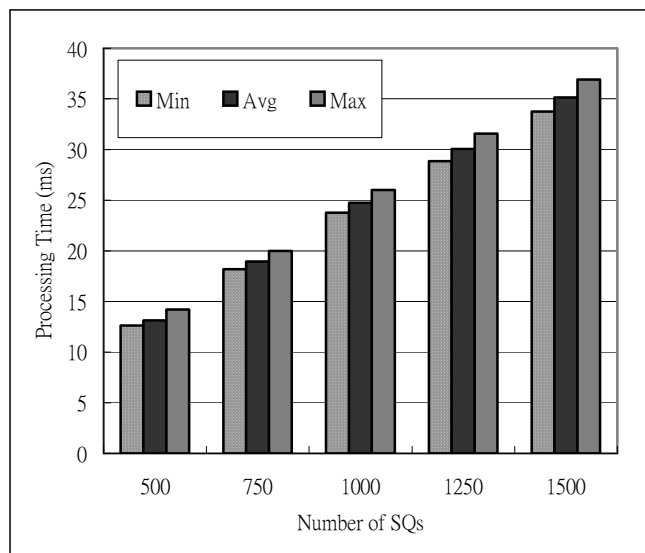


圖十：事件序列篩選系統之流程圖

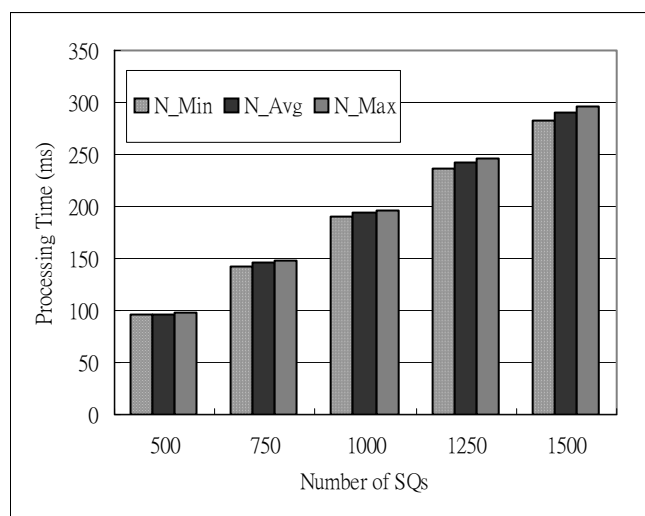
為了模擬一串流環境來進行測試，我們隨機選擇了一百首 MIDI 格式的音樂連續播放，以作為實驗所用之事件串流，這一百首歌曲一共包含了 168283 個事件，另外，我們也隨機從中選擇了一千五百個包含 24 個事件的音乐片段來作為查詢，由於我們的技術在此領域中是首

度開發，因此我們與一必須要逐步比對的暴力法做效能上的比較。

在及時性的測試之中，我們展示了處理每個事件所需的最大、最小以及平均時間，結果如圖十一與圖十二所示，我們可以發現當查詢的數量增加時，我們的方法所需耗費的時間遠比暴力法所需耗費的時間小，可以驗證我們方法所採用的多項概念與技術，的確能夠有效地減少比對所需要耗費的時間。



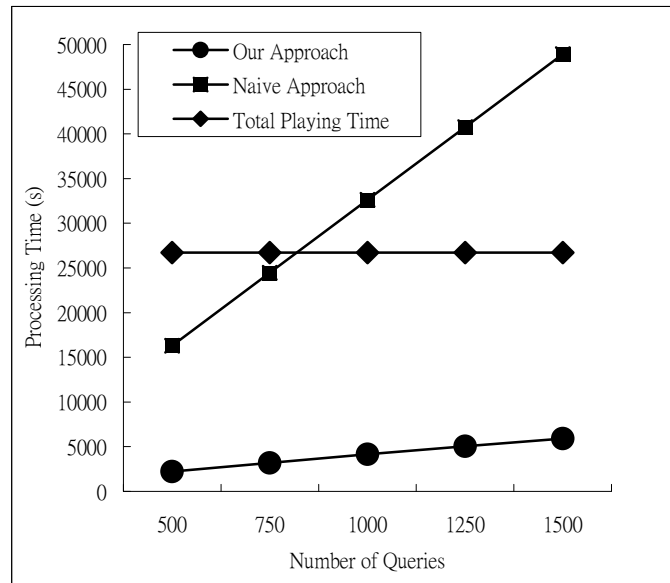
圖十一：在不同查詢數量下我們方法所需之處理時間



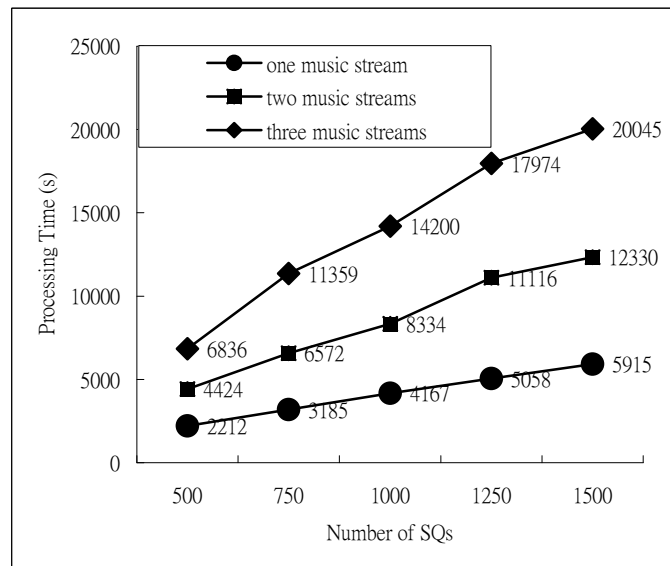
圖十二：在不同查詢數量下暴力法所需之處理時間

圖十三顯示出兩個方法的整體處理時間，在實際的情況之下，這一百首歌曲所需要播放的總時間為 7 個半小時，但是從實驗結果可以得知，當查詢的數量大於一千筆以上時，使用者可能要多等待好幾個小時以上才能夠獲取比對的結果，相較之下我們所開發的方法就可以提供令人滿意的回應時間 (response time)。

我們的方法也可以直接擴充到同時處理多條事件串流之上，結果如圖十四所示，我們比較處理一至三條事件串流所需之處理時間，結果顯示我們的方法只需要小量的額外花費辨識來自不同串流的事件即可。



圖十三：不同查詢數量下的總處理時間



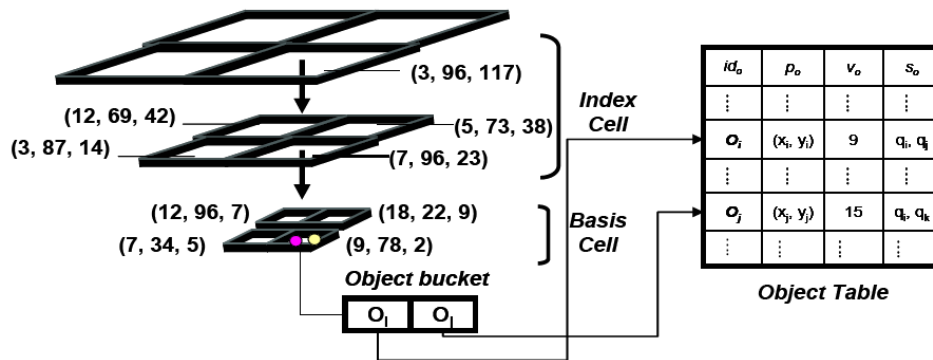
圖十四：一至三條事件串流所需之總處理時間

2. 移動物件環境下之 KNN 查詢方法

在某些應用下，資料的屬性可以分為 location-dependent 與 location-independent 兩種，例如在買鞋子時，使用者可能會同時考慮移動距離的遠近以及價格高低來決定哪些店面是可以考慮的對象，因此我們將這一類的 KNN 查詢稱為 heterogeneous KNN (HKNN) 以便與傳統只考慮 location-dependent 的 KNN 作區隔。我們的 HKNN 查詢處理方法設計了一個特殊資料結構來加速移動物件的資料紀錄，並且採用了數個全新的篩選技術來提升查詢之效率。

為了能夠有效地建立並且維護移動物件的資訊，我們設計了一個全新的資料結構：階層型聚合式網格索引架構 (hierarchical aggregate grid index)，這個索引架構由多層的網格所構成，最底層的網格由 basic cell 所構成，而其他層的網格則由 index cell 所構成，每個 index cell 都會有 link 指向它所包含的下一層 cell，圖十五為該索引架構的一個範例。每個 basic cell 會紀錄目前有哪些物件位於該 cell 所負責的範圍之內，以及該 basic cell 中的物件所擁有的最大及最小

的 location-independent 屬性的值，而 index cell 則是紀錄所有包含的 cell 中所擁有的最大與最小 attribute 的值，這些資訊將可以用來加速後續查詢處理的效能。

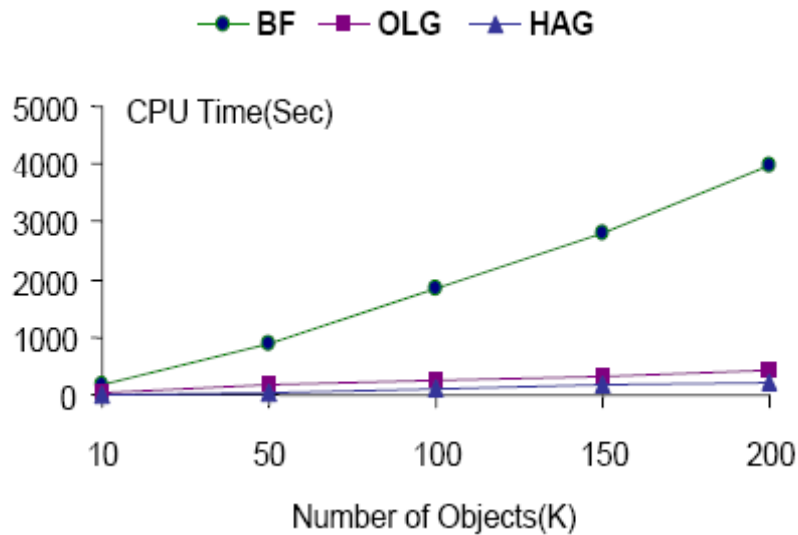


圖十五：階層型聚合式網格索引架構

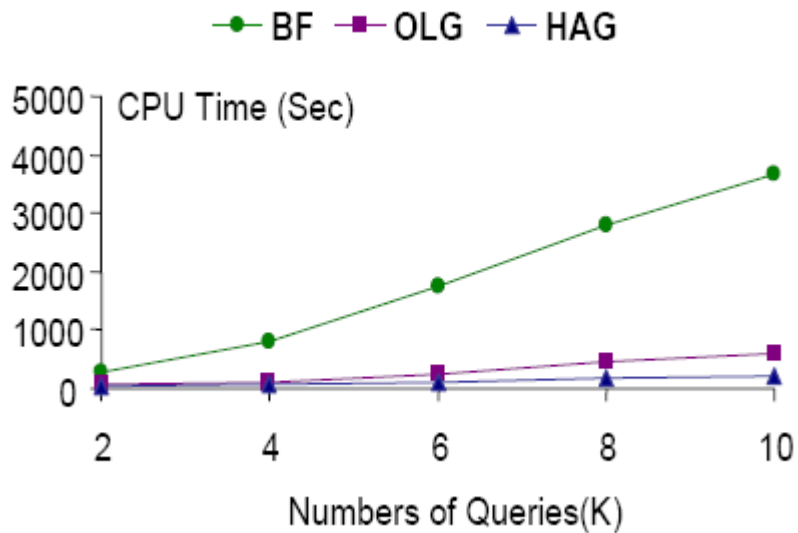
在進行查詢時可以分為兩個步驟，第一步是用傳統 KNN 的方法找出前 K 個最有可能的答案，第二步則是利用篩選機制來找出 HKNN 真正的答案：在步驟一時，為了加速查詢處理的速度，我們並不會直接利用物件本身的資訊做計算，而是先利用每一個 cell 所代表的區域以及該 cell 所包含的物件數量來作為評量物件順序的依據，直到我們沒有辦法判別物件間的順序時，才會使用物件本身的位置資訊來做計算，這樣的作法除了可以避免大量物件資訊的存取，也可以減少不必要的計算；在步驟二時，我們則是利用第一步所找出的 K 個物件作為候選者，並用擁有最小花費的物件來作為低標，檢查其餘還保有資格的物件，如果原本的 K 個物件有人被取代的話，低標會再度被降低，保有資格的物件也會再度減少，我們就利用這種反覆檢查與取代的方式來找出最後的答案，在此我們提出了 Pruning by Distance Bound 以及 Pruning by Value Bound 兩種機制來縮減需要檢查的物件數量，Pruning by Distance Bound 利用每個 cell 所代表的區域範圍來判斷在裡面的物件是否有可能擁有比目前的 K 個答案還小的花費，而 Pruning by Value Bound 則是利用每個 cell 所紀錄的各種屬性的最大最小值來判斷在裡面的物件是否有可能擁有比目前的 K 個答案還小的花費，這兩個機制都能夠有效地加速 HKNN 查詢處理的效率。

最後，延伸到連續型的 HKNN 查詢，利用前次查詢所得到的結果，我們引入了 distance influence region 概念，讓我們可以劃分出哪些 cell 中的物件變動是不會影響結果，哪些 cell 中新增或是減少物件會影響到目前的結果，而 value influence region 的概念，則是讓我們知道每個 location-independent 屬性的值可以變化的範圍大小，藉以監控哪些物件的 location-independent 屬性的更新值會影響到目前的答案，藉由這兩種 region 的劃分，我們避免了每個物件更新都必須要重新計算的窘境，也自然得以提升連續型 HKNN 查詢處理的效能。

我們將採用階層型聚合式網格索引架構的方法 (HAG) 與採用單層網格索引架構 (OLG) 以及暴力法 (BF) 作比較，結果如圖十六與圖十七所示，在不同的移動物件數量或是不同的查詢數量之下，我們的方法都比另外兩個方法的效能為優越許多。



圖十六：不同移動物件數量下三種方法的效能



圖十七：不同查詢數量下三種方法的效能

3. 交易串流環境下之區間查詢技術

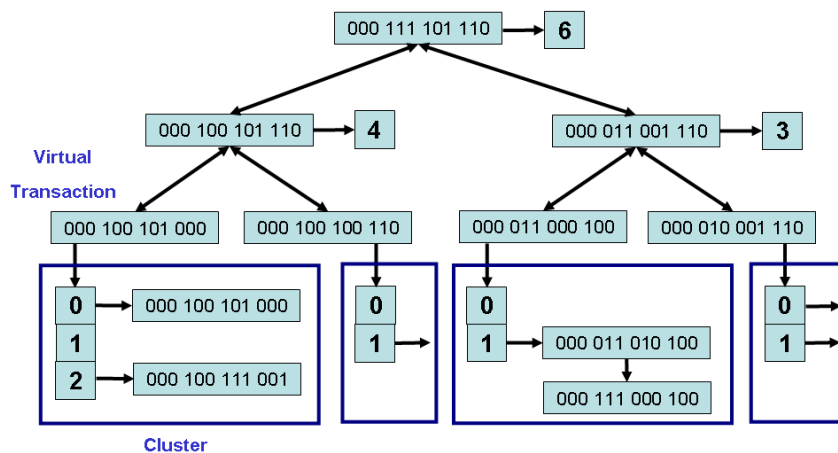
在本計畫中，我們針對在交易串流環境下進行區間查詢的問題，發展了一個全新的方法，這個方法結合了資料摘要、資料分群與分群索引的技術，來提升整體查詢的效率，在整個方法中，我們則是採用 hamming distance 來作為距離衡量的標準。

過去的針對交易型資料的查詢研究，都會利用 union transaction 來代表一群 transaction，該 union transaction 即為這些 transaction 的聯集，利用 union transaction 可以計算出查詢與該群 transaction 的距離界限，可以用來加速查詢處理的時間，然而在我們的實驗觀察中，我們發現 union transaction 常會過於鬆散，而導致距離界限的預估過為寬鬆而降低了其篩選的能力，為了解決這一個問題，我們設計了一個全新的 virtual transaction，virtual transaction 是由一群 transaction 中最常出現的數個項目所構成的，我們還額外紀錄了 r ， r 代表了這群 transaction 與 virtual transaction 的最遠距離，當一個查詢進來的時候，我們可以直接計算該查詢與 virtual transaction 的距離，加上 r 即為該群 transaction 與查詢的最大可能距離，減去 r 則為該群 transaction 與查詢的最小可能距離。

為了要能夠有效地發揮 virtual transaction 的優點，我們需要一個分群演算法，希望能夠將相似的 transaction 分群在一起，而這些 transaction 要盡量擁有共同的頻繁項目。為此，我們的演算法同時考慮頻繁項目以及分群中 transaction 的平均長度，藉以選出一個合適的 virtual transaction，在新增一個 transaction 時，則是優先把這個 transaction 加到頻估距離最近的 virtual transaction 當中，如果這個新增的 transaction 距離每個 virtual transaction 都大於系統預設的 α 值時，該 transaction 便會自己形成一個新的分群。

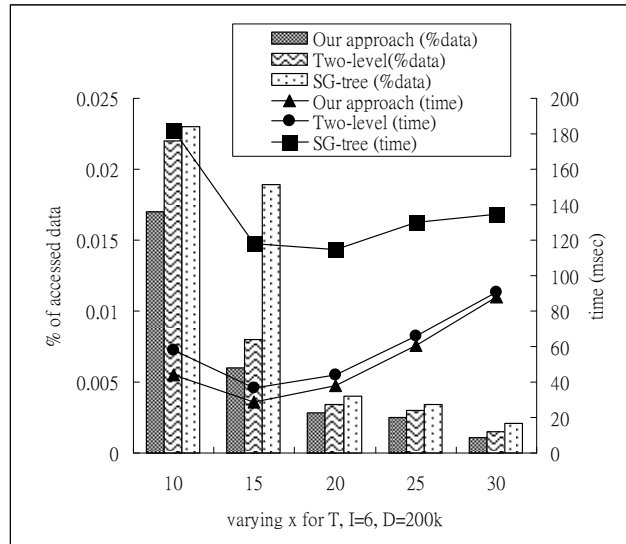
由於資料量可能十分龐大，為了避免檢查每一個分群所造成的時間耗費，我們引入了一個新的分群索引架構如圖十八所示，希望能夠透過索引架構的幫助，快速地淘汰那些不可能包含答案的分群。該索引架構是採取bottom-up的方式構建，最底層為實際存在的transaction分群，我們透過距離的計算，將兩個virtual transaction最相近的分群合併到上一層之中，並且重新計算合併之後的virtual transaction來代表這兩個分群，就這樣由下往上建立出一個類似平衡樹的索引架構。

當進行區間查詢處理時，我們會從索引架構的 root 出發，計算 virtual transaction 與查詢的評估距離是否大於查詢的容許值，如果沒有超過，則繼續往下檢查所屬的子節點的 virtual transaction，直到底層為止，如果某一節點的 virtual transaction 與查詢的評估距離已經超過容許值，則我們可以確定此節點以下所包含的所以 transaction 都不可能是答案。

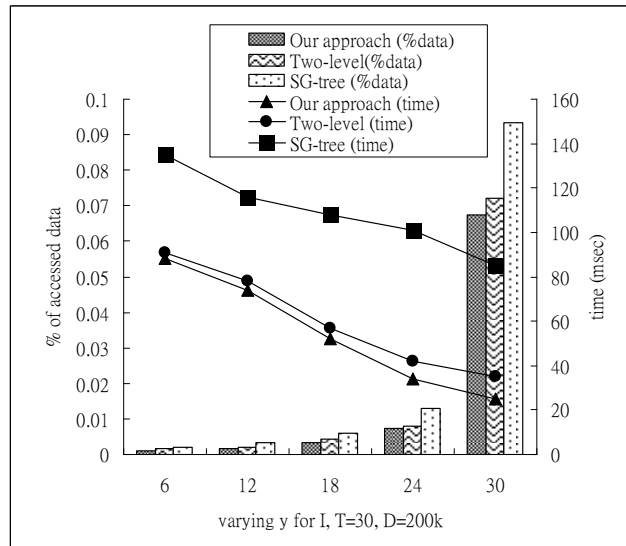


圖十八：分群索引架構

為了驗證這個方法的效率，我們將這個方法與SG-Tree以及Two-level Approach作比較，實驗資料是利用在資料探勘領域中常使用的資料產生器所提供，藉由測試不同組成的測試資料，如圖十九與圖二十所示，這個新的方法的確在不同的情況之下，都能夠提供比較好的查詢處理效率；同時，我們也測試了這個新的索引架構在更新所需的耗費，結果也顯示每 10 萬筆的資料只需要 5.5 毫秒的處理時間，所以足以應付所面對的串流環境。



圖十九：改變每筆資料所包含的平均項目數時的效能變化



圖二十：改變每筆資料的可能最大式樣時的效能變化

(三) 感測器網路之查詢處理

以下分別介紹在本項目中所開發的四項技術以及一個實作之感測系統：

1. 使用期限下之感測排程技術

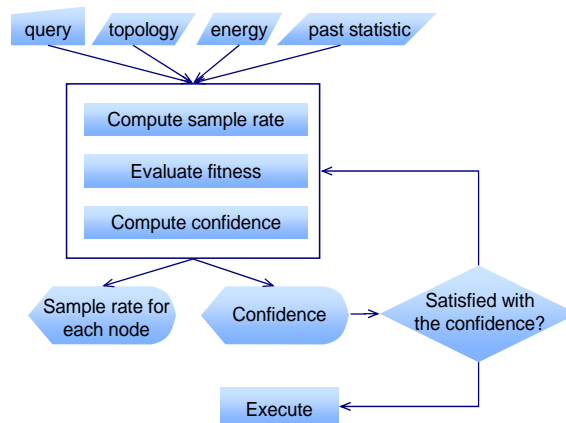
在感測器的網路拓樸結構採用樹狀結構的前提下，越是接近伺服器端的節點，必須負責傳遞越多下端子節點的封包；因此，所需消耗的能量將隨著子節點的數目而增加。其次在於針對使用者所要求的特定條件，其所需消耗能量的多寡若超出原先統計分析的估計值時，如何重新排程以求盡可能達到原先預定使用期限。為此，我們設計了一套能有效管理感測排程的查詢系統，針對各種不同的變因得出最佳的排程，並回覆使用者，根據此排程結果執行的可靠度 (confidence)，來決定此查詢的可行性，或得知應當需要如何修正，該系統架構如圖二十一所示。

本方法假設網路分布狀態和密度皆為已知，且節點均勻分佈於監測區域，原因在於均勻分佈下能平均回收各地區的資料，使得資料的代表性較高。首先我們依據過去的統計值資訊，假設資料為常態分佈 (normal distribution)，對突發事件的機率和平均耗電量做一估計，接著

保留估計期望值加上 X 倍的容許偏差範圍內之電量後，將剩餘電量依據節點數和階層數做平均分配。

假設各節點在初始時，電力皆相同的情況下，因為越上層的節點其負責傳遞的資料量相對越多，因此，上層的節點則成為整個系統執行時間的瓶頸 (bottleneck)。在期望取得的資料有區域代表性的前提下，我們希望能平均取得各個單位區域的資料，故最佳結果應是各節點回收的資料筆數差異度最小的情形，而又同時能將電量盡量耗盡，以達到最高的資料總量。因此，在保留突發事件處理的電量之後，剩餘電量會依據上列最佳解的定義來分配使用，並求出各階層節點的任務週期 (duty cycle)。而後根據統計之機率和標準差使用 Chebyshev's Inequality 來估計此一系統成功執行的可靠度，並視使用者對此可靠度滿意與否，來決定是否要增加保留給突發事件之電量，已達到更高的可靠度，或是犧牲部分可靠度來換取較高的偵測頻率 (Sample rate)，來得到較多的資料筆數。

System Model



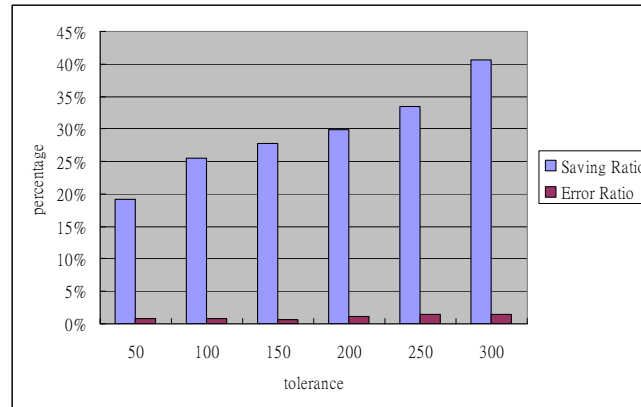
圖二十一：無線感測器感測排程技術之系統架構

2. 以趨勢分析為基礎之感測省電技術

在本研究中，其方法主要分成兩個部分：1、歷史資料趨勢分析：系統先固定一個密集性的感測時間，當收集一定資料量之後，便開始計算其週期；在得到週期之後，我們利用此週期將資料離散的切割，並把相對應時間點的資料收集在一起。我們假設週期內所有相對應時間點的資料分佈呈現常態分佈，並利用此分析得來之資料趨勢當作未來估計資料的依據。2、未來資料估計與新進資料更新：當歷史資料趨勢模型建立之後，我們將各資料模型分兩種情況來討論：a、某些過去時間點的資料分佈太過分散（資料的變異程度太大），由於該時間點上的資料將無法有效的估算未來相對應時間點的資料，所以對應到其未來時間點時，必須要實際偵測（無法省電）。b、其他過去時間點的資料分佈較為集中（資料的變異程度較小），對應到其未來時間點，我們有 p 的機率實際去偵測該筆資料（無法省電）， $1-p$ 的機率利用資料趨勢模型的期望值來回報（達成省電目的）。以上的兩種狀況，當有實際偵測資料時，該資料會加入資料趨勢模型中，達到更新資料趨勢的效果。利用 Chebyshev's Inequality 我們可以保證回傳的估計資料，在一定的信心水準之下，落在可容許的誤差範圍之內。

實驗結果如圖二十二所示，我們所提出的方法在將近 50% 的省電率中，資料錯誤率僅不

到 5%，証明了我們的系統在省電率與錯誤率之間提供了非常良好的平衡。



圖二十二：能源節省比率及資料錯誤率

3. 感測器網路之近似聚合查詢技術

現階段無線感測器網路中主要有兩種資訊傳遞方式：樹狀式資料遞送 (Tree-Based Routing) 與多路徑式資料遞送 (Multi-Path Routing)。樹狀式資料遞送方式，主要針對整個感測器網路，建構一個以主機 (Host) 為根節點的擴張樹 (Spanning Tree)，網路節點所感測或接收之資料，皆往其母節點傳送。但如前文所述，在 Zigbee 的架構下，節點間通訊封包流失率約為約百分之三十，許多節點的感測資料將因此而遺失。另一方面，多路徑式資料遞送，則將整個網路拓撲建構為以主機為終節點的 DAG (Directed Acyclic Graph)，網路節點所感測或接收之資料，皆往其上層節點傳送，相同的資料可能會被接收多次。這樣的傳遞方式，會導致相同資料重複計數 (Double Counting)。重覆計數在某些查詢下，並不會對查詢結果產生影響，例如求取網路中擁有最大溫度值的感測器編號。但是對於某些查詢，例如計算個數 count(*)，則造成錯誤的查詢結果。有趣的是，多路徑式資料遞送的方式，由於單一筆資料被接收多次，反而在 Zigbee 的架構下，擁有較佳的容錯能力 (Fault Tolerance)。

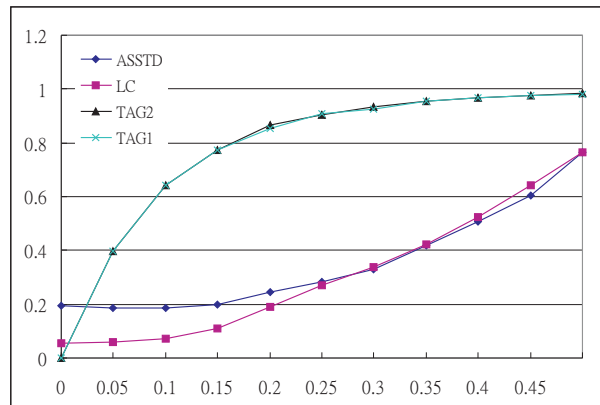
樹狀式資料遞送與多路徑式資料遞送各有其優缺點。在沒有網路錯誤發生的狀況下，樹狀式資料遞送擁有精確的查詢結果，多路徑式資料遞送則僅能提供近似 (或錯誤) 的結果。而在網路狀況不佳的條件下，樹狀式資料遞送將會遺失大部分的感測結果，而多路徑式資料遞送則保有較佳的容錯能力，並提供一個較佳的查詢結果。有鑒於此，我們將提出一個在多路徑式資料遞送方式下，有效避免聚合查詢重複計算的資料摘要結構與演算法，並提供一個有誤差保證的近似查詢結果。

多路徑式資料遞送方式產生近似結果的原因，來自於對相同的一筆感測值進行多次的重複計數 (Double Counting)。這樣的問題類似於傳統資料庫中，針對表格 (Relation) 中某個屬性 (Attribute) 進行相異值數量 (Distinct Value) 的估算問題。因此，我們引用線性計算速寫技術 (Linear Counting Technique) 來避免相同感測器被重覆讀取的問題。

線性計算速寫技術主要包含一個隨機的 (Randomized) 雜湊資料結構。線性計算速算技術主要用來估算一個多重集合 (multi-set) 中，相異值的數量。給定一個多重集合，線性計算速算技術的使用方法如下。首先，產生一個長度為 m ，初始值為零的位元陣列。同時引入一個均勻散佈且獨立的雜湊函式，該函式將給定之多重集合中的元素對應至所產生的位元陣列，並將所對應到的位址設定為一。接下來，將所有多重集合中的元素，對應至位元陣列。最後，計算位元陣列中，所有非零的位址數目。並利用非零的位址數目 (V_n)，進行相異值的

估算。相異值 \hat{n} ，可利用下列估算子 (estimator) 計算: $\hat{n} = -m * \ln(V_n)$ 。我們的估算子，可利用統計機率分析的方式，證明查詢結果擁有極高的近似保證。除此之外，我們的估算子可根據使用者所給定的允許誤差進行雜湊資料結構空間的調整，兼具查詢的準確率與節省資料結構空間使用的。

在模擬實驗中，我們使用與[CLK04]相同的實驗設定，同時實作我們所提出的方法(LC)，並以TAG[MFH02]與ASSTD[CLK04]為實驗的比較對象。在模擬的環境中我們下達Count(*)的聚合查詢，比較調整通訊錯誤(communication link failure rate)來觀察各方法的容錯能力。圖二十三為我們實驗結果，圖中橫軸為通訊錯誤率，而縱軸為平方根標準錯誤誤差(RMS)。從圖中可以看到我們的方法明顯的擁有錯誤容錯率。並且比現階段的方法擁有更高的準確度。

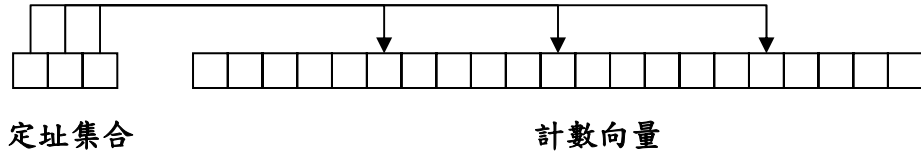


圖二十三：錯誤誤差與通訊錯誤率之關係圖

4. 以動態計數速寫為基礎之高可靠度感測資料聚合處理技術

在本計畫中，我們以多路徑式資料傳遞為基礎，設計出一個具高可靠度、高準確率與高能源效率之無線感測器資料聚合計算方法。多路徑式資料傳遞的使用，將可在使用現階段感測器網路之底層通訊協定下，大幅地提升感測器資料傳遞上的網路通訊容錯能力。然而，多路徑式資料聚合之使用，將帶來感測器資料重複計數之副作用。多路徑式資料聚合中重複計數問題之濫觴，來自於一筆部分聚合資料在網路廣播中所產生的複本，被感測器節點重複地計數。因此，如何避免相同的資料被重複地計數，成為多路徑式資料聚合計算準確率之關鍵所在。

我們引用並延伸線性計算速寫技術(Linear-Counting Sketches)來避免多路徑式資料聚合中重複計數問題。線性計算速寫技術主要包含一個隨機雜湊資料結構。給定一個多重集合，線性計算速算技術的使用方法如下。首先，配置一個長度為 m ，初始值為零的位元陣列。同時，使用一個均勻分的雜湊函式，該函式將多重集合中的元素對應至該初始值為零的位元陣列，並將所對應到的位址值設定為壹。接下來，將所有多重集合中的元素，逐一對應至位元陣列中。最後計算位元陣列中，所有位址值非零的位址數目 (V_n)，並利用下列公式估算 $\hat{n} = -m * \ln(V_n)$ ，多重集合中相異元素數量 \hat{n} 。使用線性計算速寫技術，一個多重集合中相同的元素，會被對應到相同的對應位址，將使得相同的元素不會被重複地計算。



圖二十四：動態計數速寫資料結構

由於在感測器環境的應用上，些許的估算上誤差是可被允許的，我們所設計的資料結構根據使用者所允許的誤差與誤差值的變異數，來設定資料結構長度。可以想見，此資料結構的長度應與使用者所允許的誤差、所允許的誤差值變異數，以及真正的資料聚合值有關。然而，真正的資料聚合值，即所求，在決定資料結構長度時是不知道的。因此過去常使用真正資料聚合值的上限(例如網路中的感測器數目)來決定資料結構長度。這樣的方法確實避免掉決定資料結構長度時之問題，但是卻帶來資料結構長度過長以及在離網路主機較遠之節點大多傳遞大部分值為零的位元陣列之副作用，造成大量的感測器能源消耗。因此我們提出一個動態配置資料結構長度的演算法，動態地調整資料結構的長度，以避免資料結構過長及傳遞大部分值為零的位元陣列之副作用。我們的方法進行步驟如下：

(1) 使用動態計數速寫結構表示感測器資料：首先，令所有參與資料聚合的節點 u_i ，根據其感測資料值 v_i ，配置一長度為 m_i 之動態計數速寫資料結構。其中 $m_i = \left\lceil \frac{v_i \cdot L(k, \epsilon, \delta)}{5(e^t - t - 1)} \right\rceil \cdot 5(e^t - t - 1)$ ，

詳細設定見[FC09]。一個節點的動態計數速寫資料結構 $DC(BI, CV)$ 包含一個定址索引元件 BI (Border Index)與一個計數向量 CV (Counting Vector)。圖二十四為我們所設計之動態計數速寫資料結構之概念圖。定址索引元件提供不同節點之計數向量長度索引位置，而計數向量則為一初始值為零之位元陣列。接著所有節點 u_i 將所配置之動態計數速寫資料結構之計數向量隨機均勻的設定 v_i 個元素為壹，並將定址索引元件指向 m_i 的位置。

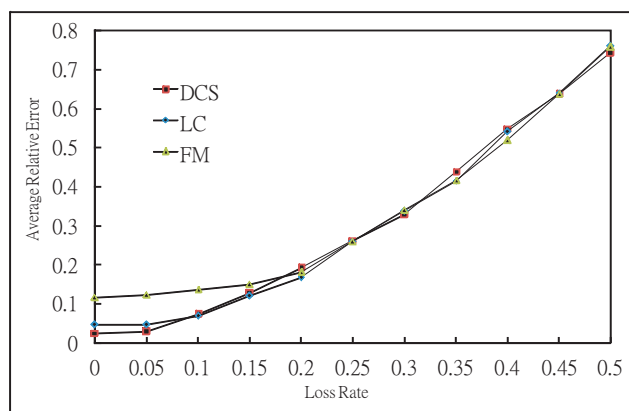
(2) 內網路動態計數速寫資料結構聚合計算：接下來，則進行內網路動態計數速寫資料結構聚合計算。該計算由網路最底層的節點開始，將其動態計數速寫資料結構廣播到其上一層的節點。收到動態計數速寫資料結構之節點，將其自己之動態計數速寫資料結構與接收到之動態計數速寫資料結構進行內網路資料聚合。當兩個動態計數速寫資料結構要進行聚合時，其聚合動作定義如下：給定兩個動態計數速寫資料結構 $DC_1 (BI_1, CV_1)$ 與 $DC_2 (BI_2, CV_2)$ ，假設 $|CV_2| \geq |CV_1|$ 。則 DC_1 與 DC_2 之總和 $DC_3 (BI_3, CV_3)$ 為(1) $BI_3 = BI_1 \cup BI_2$ 與(2) $CV_3[i] = CV_1[i] \vee CV_2[i]$, for $i = 0, \dots, |CV_1|-1$, and $CV_3[i] = CV_2[i]$, for $i = |CV_1|, \dots, |CV_2|-1$ 。接著該節點將聚合過後的動態計數速寫資料結構以此類推地往其上層節點傳送與處理，直到所有的動態計數速寫資料結構集合於網路主機節點。

(3) 近似資料聚合結果計算：待所有的動態計數速寫資料結構集合於網路主機節點後，網路主機將所有動態計數速寫資料結構聚合後產生一個最終動態計數速寫資料結構，並根據該動態計數速寫資料結構之定址索引元件與計數向量所提供的資訊利用動態計數估算子

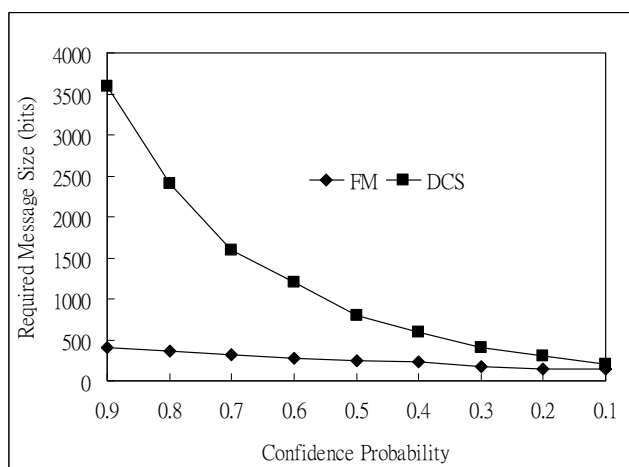
$$\hat{n} = \sum_{i=1}^{\max} \hat{v}_i = -m_{\max} \ln(V_{m_{\max}}) - \sum_{i=1}^{\max-1} m_i \ln(V_{m_i} / V_{m_{i+1}})$$

進行最終近似資料聚合結果計算，並將最終結果回傳予下達查詢之使用者。

我們實作我們所提出的方法(DCS)與LC[FC08]及FM[CLK04]來進行比較。以下簡列兩個具代表性的實驗，首先圖二十五為近似資料聚合結果準確率的比較圖表，從圖中，我們可以觀察到，在使用相同空間時，動態計數速寫資料結構提供較高準確率。另外動態計數速寫資料結構也兼具使用上的能源效率以及準確性，意即，我們使用較少的空間得到較高的準確率。另外我們調整使用者所給定的誤差需求，來觀察我們的方法與現有方法的差異，首先我們設定使用者所給定的誤差為 10%，並調整其信心水準保證來觀察兩個方法滿足使用者所給定的誤差需求時所需要的傳輸量。圖二十六為我們的實驗結果。我們可以很清楚的看到我們的方法 DCS 很明顯且優越地使用較少的傳輸量，來達成使用者所給定的誤差需求。



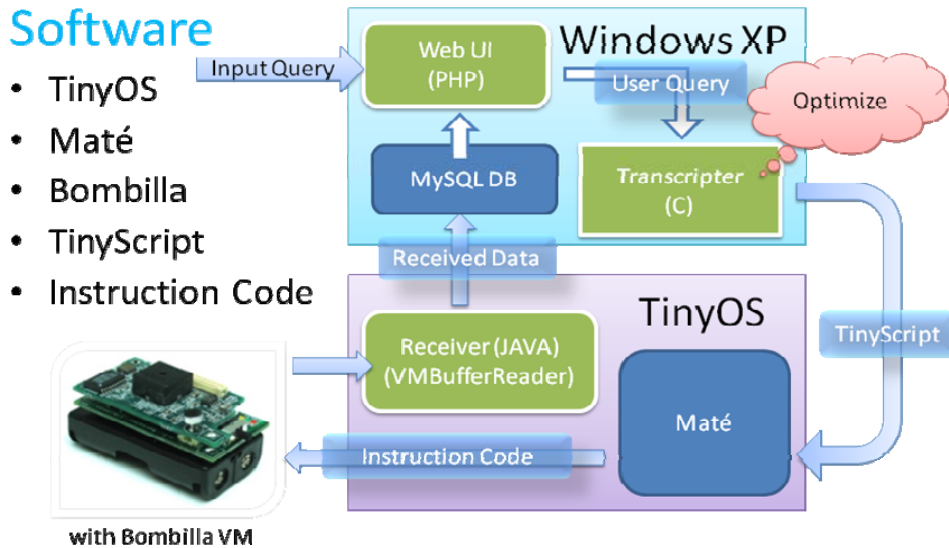
圖二十五：準確率實驗比較



圖二十六：所需空間實驗比較

5. MAKE DB 感測查詢系統

為了提供使用者親切的web-based介面，利用和SQL相似的（SQL-like）的查詢語言，來對無線感測器網路下查詢，作資料蒐集，我們擬設計一系統將底層的低階語言與上層的使用者介面分開，讓使用者不需要接觸底層架構即可利用友善介面下查詢，因而開發了MAKE DB 感測查詢系統。同時，此一系統更可提供我們在無線感測器網路相關研究上的資料蒐集。



圖二十七：MAKE DB 系統架構圖

MAKE DB的系統架構如圖二十七所示，其中所使用之相關元件分述如下：

NesC：由柏克萊大學所設計的一種專門用來開發感測器應用程式的程式語言(類似 C 語言)。這種程式語言是採取元件導向的結構，以元件來表達抽象的系統函式和硬體，而感測器程式就是組合各功能的元件來達到整體程式的目標。

<http://www.tinyos.net/dist-1.1.0/tinyos/windows/nesc-1.1.2a-1.cygwin.i386>

TinyOS：是專為無線嵌入式感測器系統所設計的作業系統，採用元件導向的結構。可依據嵌入應用，輕易增減控制執行功能，且程式碼佔量極少，有助於記憶體空間的硬體資源精省，且能夠同時執行多個要求快速回應的控制運作。TinyOS 內部是由 nesC 所寫成的各種元件集合而成。不需行程(Process)管理，不需虛擬記憶體，不需記憶體管理，採用靜態配置記憶體技術。

http://tinyos.net/windows-1_1_0.html

Maté：Maté 是 TinyOS 的一個元件，是利用 NesC 所撰寫出來的。可以想像它是一個物件，用來提供 VM(Virtual Machine) 開發。此外 Maté 還包括了一個 Java toolchain 的部分，提供我們使用一種較簡單、較高階的 scripts language 來對 TinyOS networks 進行程式的撰寫。

<http://www.cs.berkeley.edu/~pal/mate-web/rpms/mate-asvm-2.2-1.noarch.rpm>

Bombilla：Bombilla 正是用 Maté 所開發出來的一個 VM 實例。而我們目前系統所使用的 Maté，在它上面執行的正是 Bombilla 這套 VM。

TinyScript：由 Maté 所支援的一種簡單且高階的 scripts language。

Instruction Code：一種類似組合語言的 code。我們所寫的 TinyScript 程式會被 Maté 編譯成 Bombilla 能看懂的 Instruction Code。這種 Instruction Code 會被散佈(Broadcast)到感測器網路上去執行。

五、成果自評

本計畫已經順利執行完畢，並且遵循原本所提出之研究主題進行相關技術之開發，成果可分為三大主軸：資料串流的分析與監控、串流資料之連續型查詢處理以及感測器網路之查詢處理。在資料串流的分析與監控部分，分別針對資料探勘與資料統計兩個領域進行研究，以作為後續研究、監測或者是決策之用，在資料統計的領域已經寫成論文並發表於國際會議[5]，在資料探勘的領域中，針對樹狀資料探勘的技術開發已經寫成論文並發表於國際會議之上[3]，而在高頻樣型探勘方面的成果則是投到了知名國際期刊 DMKD[9]，也已經進入第二輪的審查。在串流資料之連續型查詢處理的部分，發展出數個針對不同資料類型所形成之串流的查詢處理技術，以滿足不同的應用需求，其中事件串流環境下之近似比對技術已經被著名的國際期刊 *Multimedia Systems Journal* 接受並且發表[1]，而移動物件環境下之 KNN 查詢方法也已經寫成論文發表於國際會議之上[4]，而其他成果如交易串流環境下之區間查詢技術[6]也已經投稿，這些技術都可以有效地提升查詢處理的效率，讓串流環境下的各類應用服務品質得以進一步地強化。最後，在感測器網路之查詢處理的方面，針對感測器資料之近似聚合查詢問題，第一個成果已經發表於國際會議之中[2]，後續的改良則是則是在投稿的階段之中[8]。總結本計畫之成果，所有開發出之技術都已經寫成論文，並且均進行實驗測試來驗證其效能，重要的成果都已經被著名的期刊與會議所認可，我們也將持續將其他研究成果加以發表。

1. 已經發表之學術論文

- [1] H.C. Chen, Y.H. Wu, Y.C. Soo, and A.L.P. Chen, "Continuous Query Processing over Music Streams based on Approximate Matching Mechanisms," *ACM/Springer Multimedia Systems Journal*, 2008. (SCI, EI)
- [2] Y.C. Fan and A.L.P. Chen, "Efficient and Robust Sensor Data Aggregation using Linear Counting Sketches," *Proc. IEEE International Parallel and Distributed Processing Symposium*, 2008.
- [3] M.C.E. Hsieh, Y. H. Wu, A.L. P. Chen, "Discovering Frequent Tree Patterns over Data Streams," *SIAM Conference on Data Mining*, 2006.
- [4] Y.C. Su, Y.H. Wu, and A.L.P. Chen, "Monitoring Heterogeneous Nearest Neighbors for Moving Objects Considering Location-Independent Attributes," *Proc. International Conference on Database Systems for Advanced Applications*.
- [5] T. C. Wu and Arbee L.P. Chen, "Maintaining Moving Sums over Data Streams," *Proc. International Conference on Advanced Data Mining and Applications*, 2006.

2. 投稿中之學術論文

- [6] H.C. Chen, Y.H. Wu, and A.L.P. Chen, "Similarity Search over Transactions in a Dynamic

Environment,” *Submitted for publication.*

- [7] Y. C. Fan and A.L.P. Chen, “Exploring Common Evaluation for Efficient Processing Multiple Queries in Sensor Databases,” *Submitted for publication.*
- [8] Y.C. Fan and A.L.P. Chen, “Efficient and Robust Sensor Data Aggregation using Dynamic Counting Sketches,” *Submitted for publication.*
- [9] E.T. Wang and A.L.P. Chen, “A Novel Hash-based Approach for Mining Frequent Itemsets over Data Streams Optimizing Memory Space Utilization,” *Submitted for publication.*
- [10] E.T. Wang and A.L.P. Chen, “Continuously Maintaining a Global Synopsis for Distributed Finding Frequent Itemsets over Data Streams,” *Submitted for publication.*

3. 碩士畢業論文

- [11] C.S. Chen and A.L.P. Chen, “Lifetime-based Acquisition Scheduling for Wireless Sensor Networks,” National Tsing Hua University, 2007.
- [12] G.R. Lin and A.L.P. Chen, “Adaptive Power-Saving Techniques for Wireless Sensor Networks based on Incremental Data Trend Analysis,” National Tsing Hua University, 2007.
- [13] J. Hsieh and A.L.P. Chen, “An Efficient Optimization Strategy for Query Execution over Streaming Sources,” National Tsing Hua University, 2006.
- [14] Y.K. Lu and A.L.P. Chen, “Mining Closed Frequent Subtrees over Data Streams,” National Tsing Hua University, 2006.
- [15] Y. C. Fan and A.L.P. Chen, “Sharing Aggregate Computations for Processing Multiple Regional Queries in Sensor Networks,” National Tsing Hua University, 2008.

參考文獻

- [AG90] S. Altschul, W. Gish, W. Miller, E.W. Myers, and D.J. Lipman, "Basic Local Alignment Search Tool," *Journal of Molecular Biology*, 1990.
- [AH00] R. Avnur and J.M. Hellerstein, "Eddies: Continuous Adaptive Query Processing," *Proc. ACM SIGMOD International Conference on Management Data*, 2000.
- [AS94] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules in Large Databases," *Proc. International Conference on Very Large Databases*, pp. 487-499, 1994.
- [BO03] B. Babcock and C. Olston, "Distributed top-k monitoring," *Proc. ACM SIGMOD Conference on Management of Data*, pp. 28-39, 2003.
- [CC02] D. Carney, U. Cetintemel, M. Cherniack, C. Convey, S. Lee, G. Seidman, M. Stonebraker, N. Tatbul, and S. Zdonik, "Monitoring Streams - A New Class of Data Management Applications," *Proc. International Conference on Very Large Data Bases*, 2002.
- [CCF02] M. Charikar, K. Chen, and M. Farach-Colton, "Finding Frequent Items in Data Streams," *Proc. International Colloquium on Automata, Languages and Programming (ICALP'02)*, Málaga, Spain, July 2002, pp. 693-703.
- [CD00] J. Chen, D.J. DeWitt, F. Tian, and Y. Wang, "NiagraCQ: A scalable continuous query system for internet databases," *Proc. ACM SIGMOD International Conference on Management of Data*, 2000.
- [CE00] M. Clausen, R. Engelbrecht, et al., "PROMS: A Web-based Tool for Searching in Polyphonic Music," *Proc. International Symposium on Music Information Retrieval*, 2000.
- [CF02] S. Chandrasekaran and M.J. Franklin, "Streaming Queries over Streaming Data," *Proc. International Conference on Very Large Data Bases*, 2002.
- [CGM05] G. Cormode, M. Garofalakis, S. Muthukrishnan, and R. Rastogi, "Holistic Aggregates in a Networked World: Distributed Tracking of Approximate Quantiles," *Proc. ACM SIGMOD Conference on Management of Data*, pp. 25-36, 2003.
- [CL03] J. H. Chang and W. S. Lee, "Finding Recent Frequent Itemsets Adaptively over Online Data Streams," *Proc. ACM SIGKDD International Conference on Knowledge Discovery in Databases and Data Mining (KDD'03)*, Washington, DC, USA, August 2003, pp. 487-492.
- [CLK04] J. Considine, F. Li, G. Kollios, and J. Byers, "Approximate aggregation techniques for sensor databases," *Proc. IEEE Conference on Data Engineering*, pp. 449-460, 2004.
- [CM03] G. Cormode and S. Muthukrishnan, "What's Hot and What's Not: Tracking Most Frequent Items Dynamically," *Proc. ACM SIGMOD-SIGACT- SIGART Symposium on Principles of Database Systems (PODS2003)*, San Diego, CA, June 2003, pp. 296-306.

- [CM05] G. Cormode and S. Muthukrishnan, "An improved data stream summary: the count-min sketch and its applications," *J. Algorithms* 55(1): 58-75
- [CMZ06] G. Cormode, S. Muthukrishnan, and W. Zhuang, "What's Different: Distributed, Continuous Monitoring of Duplicate-Resilient Aggregates on Data Streams," *Proc. IEEE Conference on Data Engineering*, pp. 57-66, 2006.
- [CMZ07] G. Cormode, S. Muthukrishnan, and W. Zhuang, "Conquering the Divide: Continuous Clustering of Distributed Data Streams," *Proc. IEEE Conference on Data Engineering*, pp. 1036-1045, 2007.
- [CP06] J. Y. Chen, G. Pandurangan, and D. Xu, "Robust computation of aggregates in wireless sensor networks: distributed randomized algorithms and analysis," *IEEE Transaction on Parallel and Distributed System*, vol. 17, no. 9, pages 987-1000, 2006.
- [CWY04] Y. Chi, H. Wang, P. S. Yu, and R. R. Muntz, "Moment: Maintaining Closed Frequent Itemsets over a Stream Sliding Window," *Proc. IEEE Conference on Data Mining*, pp. 59-66, 2004.
- [DGG04] A. Das, S. Gangule, M. Garofalakis, and R. Rastogi, "Distributed Set-Expression Cardinality Estimation," *Proc. International Conference on Very Large Data Bases*, pp. 312-323, 2004.
- [DLM02] E. Demaine, A. Lopez-Ortiz, and J. I. Munro, "Frequency Estimation of Internet Packet Streams with Limited Space," *Proc. European Symposium on Algorithms (ESA2002)*, Rome, Italy, September 2002, pp. 348-360.
- [Dov99] M. J. Dovey, "An Algorithm for Locating Polyphonic Phrases within a Polyphonic Musical Piece," *Proc. AISB Symposium on Musical Creativity*, 1999.
- [FC08] Y. C. Fan and A. L. P. Chen, "Efficient and robust sensor data aggregation using linear counting sketches," *Proc. IEEE Symposium on Parallel and Distributed Processing*, pages 1-12, 2008.
- [FM85] P. Flajolet and G. N. Martin, "Probabilistic counting algorithms for database applications. *Journal of Computer and System Science*," pp. 31, 1985.
- [FS82] M. J. Fischer and S. L. Salzberg, "Finding A Majority among N Votes: Solution to Problem 81-5," *Journal of Algorithm*, Vol. 3, Is. 4, 1982, pp. 362-380.
- [GDD03] L. Golab, D. DeHaan, E. D. Demaine, A. López-Ortiz, and J. I. Munro, "Identifying Frequent Items in Sliding Windows over On-Line Packet Streams," *Proc. ACM SIGMOD Conference. on Management of Data*, pp. 173-178, 2003.
- [GHP03] C. Giannella, J. Han, J. Pei, X. Ayn, and P. S. Yu, "Mining Frequent Patterns in Data Streams at Multiple Time Granularities," *Next Generation Data Mining*, 2003, pp. 191-212.
- [GO03] L. Golab and M.T. Özsu, "Issues in data stream management," *ACM SIGMOD Record*, 2003.
- [HF00] J. Hellerstein, M. Franklin, S. Chandrasekaran, A. Deshpande, K. Hildrum, S. Madden, V. Raman, and M.A. Shah, "Adaptive query processing: Technology in evolution," *IEEE Data Engineering Bulletin*, 2000.

- [HP00] J. Han, J. Pei, and Y. Yin, "Mining Frequent Patterns without Candidate Generation," *Proc. of the ACM SIGMOD Conf. on Management of Data*, pp. 1-12, 2000.
- [JA05] R. Jin and G. Agrawal, "An Algorithm for In-Core Frequent Itemset Mining on Streaming Data," *Proc. IEEE International Conference on Data Mining (ICDM'05)*, Houston, Texas, USA, November 2005, pp. 210-217.
- [JQS03] C. Jin, W. Qian, C. Sha, J. X. Yu, and A. Zhou, "Dynamically Maintaining Frequent Items Over A Data Stream," *Proc. ACM International Conference on Information and Knowledge Management (CIKM'03)*, New Orleans, LA, USA, November 2003, pp. 287-294.
- [KCR06] R. Keralapura, G. Cormode, and J. Ramanirtham, "Communication-Efficient Distributed Monitoring of Thresholded Counts," *Proc. ACM SIGMOD Conference on Management of Data*, pp.289-300, 2006.
- [KN03] J. Kang, J.F. Naughton, and S.D. Viglas, "Evaluating Window Joins over Unbounded Streams," *Proc. IEEE International Conference on Data Engineering*, 2003.
- [KPS03] R. M. Karp, C. H. Papadimitriou, and S. Shenker, "A Simple Algorithm for Finding Frequent Elements in Streams and Bags," *ACM Transactions on Database Systems*, Vol. 28, Is. 1, 2003, pp. 51-55.
- [KS01] T. Kahveci and A.K. Singh, "An Efficient Index Structure for String Databases," *Proc. International Conference on Very Large Data Bases*, 2001.
- [LCW05] C. H. Lin, D. Y. Chiu, Y. H. Wu, and A. L. P. Chen, "Mining Frequent Itemsets from Data Streams with a Time-Sensitive Sliding Window," *Proc. SIAM International Conference on Data Mining*, 2005.
- [LK06] C. K. S. Leung and Q. Khan, "DSTree: A Tree Structure for the Mining of Frequent Sets from Data Streams," *Proc. IEEE Conference on Data Mining*, pp. 928-932, 2006.
- [LL05] D. Lee and W. Lee, "Finding Maximal Frequent Itemsets over Online Data Streams Adaptively," *Proc. IEEE International Conference on Data Mining (ICDM'05)*, Houston, Texas, USA, November 2005,, pp. 266-273.
- [LP00] K. Lemstrom and S. Perttu, "SEMEX – An Efficient Music Retrieval Prototype," *Proc. International Symposium on Music Information Retrieval*, 2000
- [MFH02] S. Madden, M. J. Franklin, and J. M. Hellerstein, and W. Hong. "TAG: a tiny aggregation service for ad-hoc sensor networks." *Proc. Symposium on Operating System Design and Implementation*, pages 131-146, 2002.
- [MM02] G. S. Manku and R. Motwani, "Approximate Frequency Counts over data Streams," *Proc. International Conference on Very Large Databases (VLDB2002)*, Hong Kong, China, August 2002, pp. 346-357.
- [MN05] A. Manjhi, S. Nath, and P. B. Gibbons, "Tributaries and Deltas: efficient and robust aggregation in sensor network streams," *Proc. ACM SIGMOD Conference on Management of Data*, pp. 287-298, 2005.
- [MS02] S. Madden, M. Shah, J.M. Hellerstein, and V. Raman, "Continuous Adaptive Continuous Queries over Streams," *Proc. ACM SIGMOD International Conference on*

Management Data, 2002.

- [MW03] R. Motwani, J. Widom, A. Arasu, B. Babcock, S. Babu, M. Datar, G. Manku, C. Olston, J. Rosenstein, and R. Varma, "Query processing, approximation, and resource management in a data stream management system," *Proc. Biennial Conference on Innovative Data Systems Research*, 2003.
- [NG04] S. Nath, P. B. Gibbons, S. Seshan, and Z. R. Anderson, "Synopsis diffusion for robust aggregation in sensor network," *Proc. ACM Conferemce on Embedded Networked Sensor System*, pp. 250-262, 2004.
- [PL98] W.R. Pearson and D.J. Lipman, "Improved tools for Biological Sequence Comparison," *Proc. National Academy of Sciences*, 1998.
- [SB04] N. Shrivastava, C. Buragohain, D. Agrawal, and S. Suri, "Medians and beyond: new aggregation techniques for sensor networks," *Proc. ACM Conference on Embedded Networked Sensor System*, page 239-249, 2004.
- [YCL04] J. X. Yu, Z. Chong, H. Lu, and A. Zhou, "False Positive or False Negative: Mining Frequent Itemsets form High Speed Transactional Data Streams," *Proc. International Conference on Very Large Databases (VLDB2004)*, Toronto, Canada, August 2004, pp. 204-215.