

南 華 大 學  
資 訊 管 理 研 究 所  
碩 士 論 文

使用一個有效的演算法來解決無線廣播資料配置問題  
An efficient Algorithm for Data Allocation on Wireless  
Broadcast Channel



研 究 生：賴 嘉 恩

指 導 教 授：吳 光 閔 博 士

中 華 民 國 九 十 三 年 六 月

南 華 大 學  
資 訊 管 理 學 系 ( 所 )  
碩 士 學 位 論 文

使用一個有效的演算法來解決無線廣播資料配置問題  
An efficient Algorithm for Data Allocation on Wireless Broadcast Channel

研究生： 賴壽恩

經考試合格特此證明

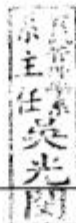
口試委員： \_\_\_\_\_

蔡德讓

周志順

指導教授： 吳光閣

所 長： \_\_\_\_\_



口試日期：中華民國 九十三年 六 月 十八 日

南華大學碩士班研究生

論文指導教授推薦函

資訊管理系碩士班 賴嘉恩 君所提之論文  
使用一個有效的演算法來解決無線廣播資料配  
置問題

係由本人指導撰述，同意提付審查。

指導教授

吳光鳴

93年6月21日

# 誌 謝

本論文的撰寫以及完成，首先要感謝的是我的神，在家庭與課業以及工作的兼顧上，祂使我無後顧之憂，尤其在論文在撰寫以及實驗程式的寫作過程之中，使我靈感泉源不絕。

也非常感謝我的指導教授吳光閔老師，在論文的編纂以及實驗的過程之中，時常提供寶貴的修正意見以及建議，建立我的寫作觀念以及正確實驗邏輯，真是由衷的感謝教授的協助與教導。並感謝各位口試委員的建議與指正，使本篇的實驗架構以及論文研究能夠更加嚴謹與完整。

在這兩年的學習期間，爸爸、媽媽、乃鳳老婆在照顧小孩上的充分支援，讓我無後顧之憂得以完成學業，在此特別獻上萬分的感謝給我最愛的家人。在此也感謝同學鳳玲、侑柯、奎佑、建億以及學妹美倫的鼓勵與幫助，使我在論文的寫作以及實驗的過程之中，得以順利完成。

# 使用一個有效的演算法來解決無線廣播資料配置問題

學生：賴嘉恩

指導教授：吳光閔

南 華 大 學 資 訊 管 理 學 系 碩 士 班

## 摘 要

無線廣播是現代行動通信不可或缺的資訊取得來源。為要減少使用者取得資料的時間，本篇提出一個運用在廣播頻道中，資料分配的演算法則，有效降低廣播循環中，使用者查詢資料的總存取時間。在我們的模擬實驗中，我們所使用的演算方法與 *Yon Dohn Chung* 和 *Myoung Ho Kim* 所提出的演算方法比較後，總存取時間提昇約 23%。由實驗結果證明我們的方法是有效率的。

關鍵字：無線廣播，廣播排程

# An efficient Algorithm for Data Allocation on Wireless Broadcast Channel

Student : Chia-En Lai.

Advisors : Dr. Guang-Ming Wu.

Department of Information Management  
The M.B.A Program  
Nan-Hua University

## ABSTRACT

The wireless broadcasting is the indispensable information origin in modern mobile communication. For reason to reduce the broadcasting user access time, we proposed an algorithm for data allocation on wireless broadcast channel. It is effectively reduce the total user access time in the broadcast circulation. We set up an experiment to compare our algorithm with Yon Dohn Chung and Myoung Ho Kim's algorithm, the total access time promote approximately 23%. The experimental result proves our algorithm is effective.

Key Words: wireless broadcast, broadcast schedule

# 目 錄

書名頁.....	i
授權書.....	ii
推薦函.....	iii
論文口試合格證明.....	iv
致謝.....	v
中文摘要.....	vi
英文摘要.....	vii
目錄.....	viii
圖目錄.....	x
表目錄.....	ix
第一章 序 論.....	1
第一節 研究背景與動機.....	1
第二節 研究目標與限制.....	8
第三節 論文架構.....	13
第二章 問題描述.....	14
第三章 我們的演算法.....	19
第一節 初步廣播資料處理.....	19
第二節 廣播資料子群組化.....	20
第三節 廣播資料權重設定.....	22
第四節 廣播子群組內資料排序.....	25
第五節 排程演算法及其參數設定.....	26
第四章 實驗結果與討論.....	28
第一節 模擬環境.....	28
第二節 實驗資料產生.....	29
第三節 廣播效能比較.....	30
第五章 結論與未來研究方向.....	42
參 考 文 獻.....	43

# 圖 目 錄

圖 1 廣播運作環境圖.....	4
圖 2 廣播架構圖.....	15
圖 3 單一查詢於無線廣播中的呈現.....	16
圖 4 初步廣播資料物件順序配置圖.....	20
圖 5 子群組切割圖.....	20
圖 6 子群組權重設定.....	22
圖 7 權重計算及子群組內資料物件排序法.....	25
圖 8 均勻分布中改變查詢資料物件數， <i>GREEDY</i> 與 <i>EA</i> 的 <i>TQD</i> 改變情形.....	31
圖 9 均勻分布中每一 <i>P</i> 值子群組切割排序循環運算次數 <i>s</i> 之變換情形.....	32
圖 10 均勻分布中改變查詢使用者數量大小， <i>GREEDY</i> 與 <i>EA</i> 的 <i>TQD</i> 改變情形.....	33
圖 11 均勻分布中改變查詢選擇率， <i>GREEDY</i> 與 <i>EA</i> 的 <i>TQD</i> 改變情形.....	34
圖 12 常態分布改變查詢資料物件數， <i>GREEDY</i> 與 <i>EA</i> 的 <i>TQD</i> 改變情形.....	35
圖 13 常態分布改變查詢使用者數量大小， <i>GREEDY</i> 與 <i>EA</i> 的 <i>TQD</i> 改變情形.....	36
圖 14 常態分布改變查詢選擇率， <i>GREEDY</i> 與 <i>EA</i> 的 <i>TQD</i> 改變情形.....	37
圖 15 循環參數設定改變， <i>TQD</i> 的改變情形.....	39
圖 16 循環參數設定改變且資料庫物件數變為 300 時， <i>TQD</i> 的改變情形.....	39
圖 17 循環參數設定改變且資料庫物件數變為 500 時， <i>TQD</i> 的改變情形.....	39
圖 18 循環參數設定改變且資料庫物件數變為 1000 時， <i>TQD</i> 的改變情形.....	40
圖 19 循環參數設定改變且使用者查詢選擇率變大時， <i>TQD</i> 的改變情形.....	40
圖 20 循環參數設定改變資料庫物件數變為 1000，且使用者查詢選擇率變大時， <i>TQD</i> 的改變情形.....	40



# 表 目 錄

表 1 符號定義.....	15
表 2 資料物件權重計算.....	24
表 3 實驗資料條設定數據.....	29
表 4 均勻分布中改變查詢資料物件數的實驗數據.....	31
表 5 均勻分布中改變查詢使用者數量的實驗數據.....	33
表 6 均勻分布中改變查詢使用者查需選擇率的實驗數據.....	34
表 7 常態分布中改變查詢資料物件數的實驗數據.....	35
表 6 常態分布改變查詢使用者數量的實驗數據.....	36
表 8 常態分布改變查詢使用者查詢選擇率的實驗數據.....	37
表 9 相關循環參數設定的實驗數據.....	38

# 第一章 序 論

## 第一節 研究背景與動機

近幾年來科技不斷的進步，人們由口耳相傳交換訊息進而信件的傳遞來傳送訊息，再則網路的出現，使得人們在使用資訊的交換運用電子郵件更為便捷，試想人們會因此而滿足嗎？當然不會。科技來自於人性，當人們擁有方便快捷的網路系統，卻不想被網路所束縛，因此無線網路就此而產生。無線網路不僅將人們的生活帶進一個方便的空間中，而這樣的資訊科技也不斷的應用於現實生活之中。像無線廣播（Wireless）就是一個例子。無線廣播是我們平常所會碰到的媒體，而這一個媒體也將人們的生活密切的結合。它的出現滿足了人們不管身在何處，都可以馬上獲得想要的資訊，甚至能夠立即與公司的資料庫聯繫，獲得相關的訊息以利於職場的瞬息萬變。

我們以往的網路架構是運用主從式架構的模式在運作，而這樣的主從式架構有著嚴重的缺陷，因為主從式架構模式網路的運作，它的資料傳送是使用單點傳送（unicast）的方式進行資料傳輸，當使用者需要資料時則發出一個要求到伺服器端，伺服器端再回送其所要求的資訊，但網路上大量使用者的要求出現時，伺服器端為了要處理大量的使用者需求而負載過重，導致伺服器端無法處理目前的所有需求。然而廣播電信通訊、行動計算、廣播系統的使用，運用擁有大量有用資訊的資料中心或是分散式資料庫，進行資料的廣播工作，提供

了資料查詢服務。而這些大量的需求則經由有效率的資料廣播分配，適切的提供查詢者相關資訊，滿足了各種使用者不同的需求。

在日常生活之中人們運用行動電話（Mobile Phone）接收簡訊，已經是相當平常的事情，但是相關的生活簡訊已經不能夠滿足一般民眾的需求。因此如何滿足特定族群的需求，進行系統的即時播放，已是通訊廠商吸引客戶的方式。例如股市瞬息萬變的資訊，需要快速的傳送資訊到投身股票市場的人們，投身股市的人何其多，想要將這些隨時在波動的股市店頭資料即時的傳送給大量的使用者是需要廣播機制來操控，而市面上就有類似的產品，一個小型的接收器，將需要的各類股的資訊設定好，接收器會定時運用系統化的方式進行目前股價的更新，賦予使用者極為方便的資訊接收方式。而先進的衛星導航系統也是運用廣播的方式進行資料的傳送，此種先進的系統已被廣泛的運用在交通資訊管理系統上，例如飛機的飛行航線導引、汽車行車路線導引或是觀光旅遊導覽位置參考。

當運用廣播方式將資料傳遞至使用者的事務日漸增加，有鑑於以往人們運用網際網路進行資料搜尋，而資料庫伺服器運用單點傳送機制將資料傳遞至使用者端已經無法滿足越來越多的使用者即時性的需求，譬如網際網路的資料查詢功能，網頁的查詢已經跳脫以往的資料查詢模式，而是運用資訊累積進行熱門資料的播送，將使用者所需的資訊傳遞給使用端。因此廣播系統被廣泛的使用，即時廣播大眾所

需求的資訊在廣播頻道中，讓大多數的使用者能夠利用監聽（monitor）的方式在頻道之中得到自己所想要的資料，因而大大的降低資訊中心重複傳送相同資訊給不同使用者的動作。而我們將這樣的方式使用於生活中相關無線資訊的傳送上，例如：將每日重點新聞或是每日娛樂快報傳遞給不同需求的使用者端 等等，能有效的將熱門的資訊放置於廣播媒介上，讓大量使用者需求透過這種廣播的方式獲得想要的資訊。

在無線廣播的環境之中，包含了大量的使用者端，以及用來存放使用者需求的循環佇列（Request loop queue），與使用相關廣播排程法則（Schedule Rule）的廣播排程伺服器（Schedule Server），並負責擔任廣播媒介的廣播器材。圖 1 顯示廣播排程伺服器收集許多的使用者端的查詢資料，並將資料查詢頻率運用排程演算法將資料物件進行資料排序最佳化，形成特定的廣播循環序列，藉著廣播系統以及器材進行重複性的資料廣播工作，而使用者端則運用監聽的方式，監聽循環播放的資料，再針對所需求的資料進行資料擷取的動作，如此進行週而復始。

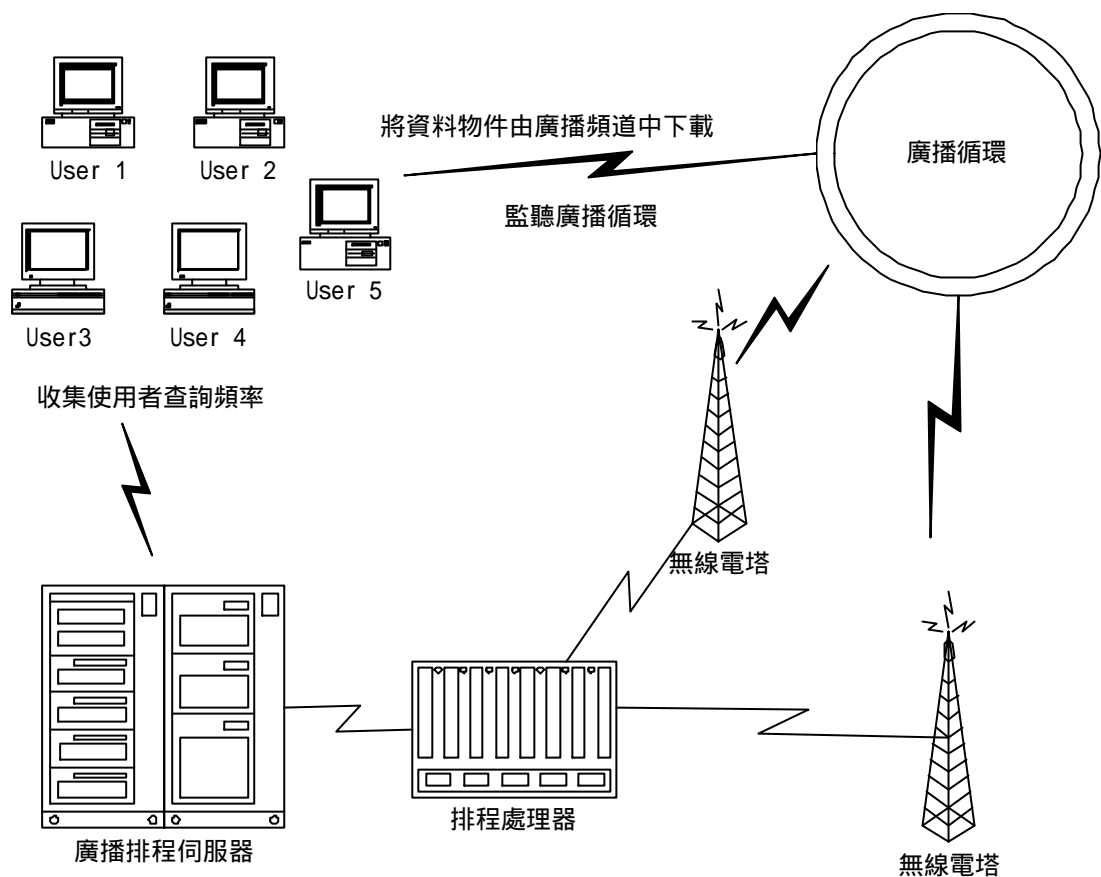


圖 1 廣播運作環境圖

這樣的廣播環境能有效提供使用者較熱門的需求資料，因著一台負責進行資料廣播邏輯演算排程的伺服器，依照歷史紀錄中使用者查詢資料種類頻率，經由資料頻率邏輯運算排程技術排列後，呈現一組資料物件循環的資料物件廣播序列 (Data object broadcast cycle)，而後將此序列依次序的進行廣播頻道的廣播。相對而言，資料物件在使用者之間的被監聽以及擷取，即如何運用排程演算法有效縮短資料物件間距，加速使用者端擷取資料物件的讀取時間 (Access time) 就顯得相當重要了。

在廣播的環境之中，我們稱存在於廣播頻道中的資料為資料物件

(Data Object) 或是稱之為資料桶 (data bucket), 經由廣播系統分析後, 決定一組序列, 依序地重複播送到廣播頻道 (Broadcast Channel) 上, 而需要這些資料的接收端皆經由持續監聽廣播頻道中的資料物件, 由其中得以擷取相關所需的資訊。

運用在無線網路、行動運算與廣播的相關領域, 其中主要區分兩種無線廣播模式, 一為推模式 (Push mode), 另一為拉模式 (Pull mode) [1], Push mode 主要是由廣播服務系統將需廣播的資料進行資料頻率的累積, 統計相關歷史資料, 依照各個資料物件的擷取頻率 (Access frequency) 進行相關演算排序後, 產生一份廣播排程資料, 在一個固定時間內, 將這組廣播排程資料有規律的放置於廣播頻道之中, 形成一組廣播循環 (Broadcast Cycle), 而使用者則進行聆聽的動作, 進行相關查詢資料的接收; Pull mode 也稱為要求式廣播 (On-demand Broadcast), 廣播系統於當下蒐集使用者所提出的需求之後, 各項資訊種類進行演算法的演算之後, 則將演算法所認定優先廣播的資料物件群放置到廣播頻道上。而將這兩種方法在使用時機上如何取得平衡點, 在無線環境的資料廣播這個領域是很重要的一個議題[9][15], 這類的廣播技術會加入一個判斷函式, 決定在廣播頻道負載輕的時候使用 Pull mode 的演算法, 在頻道負載重的時候使用 Push mode 的演算法, 使用此種混合方式的廣播技術, 可以有效互補兩種廣播模式的缺點, [9]之中更提及如何在多個頻道進行資料廣播時, 運用相關

演算方式進行 Pull mode 以及 Push mode 使用時機的判斷，加速資料在進行多頻道廣播時，廣播模式混用效率的增加。一般在進行廣播時，如果單單使用 Pull mode 模式，當系統遇到使用者大量出現的情況時，往往會在單位時間內無法立即運用演算法進行廣播資訊種類的判斷，因而會產生部分廣播頻道中時間區塊（time slot）是閒置未被使用的狀態，而產生網路廣播頻寬（bandwidth）未完全運用，若於當下使用者大量出現時轉換為 Push mode，則可運用其優點，只要決定由歷史資料庫所產生的廣播排程進行廣播，則不會被使用者的需求數量多寡所影響，而網路頻寬閒置的問題即不存在。另一方面，若單獨使用 Push mode 模式進行廣播，在某些使用者單獨要求的冷門資料（Cold data），會因為這些資料在歷史資料庫出現的機率並不高，因此排進廣播排程的機會並不高，甚至不會排入排程之中，使得冷門資料的用戶端苦等資料而毫無下文，則需要經由 Pull mode 要求模式進行資料的查詢，若經由混和模式的使用，在頻道負載重獲使用者數量過多的時候使用 Push mode，而在負載輕且用戶端少的情況之下使用 Pull mode，則能夠更有效率的滿足提出各種需求的使用者。

在無線通訊相關研究議題之中，用來評估一廣播機制的指標主要有兩種：一為讀取時間（access time），從客戶端開始下載資料至拿到所要資料所花費的時間；二為調整時間（tuning time），處於活動模式時之所有時間總和。讀取時間的長短是最主要影響行動客戶是否能快

速拿到想要的資料，一般來說此類相關研究[1][5][7][16]乃是藉由調整資料存取頻率來安排資料物件廣播時之順序，而使得讀取時間能夠縮到最小。另一評估指標值調整時間之多寡，最主要影響行動客戶端能源的消耗，此類相關的研究運用「索引技術」[8][10][11][12][13]將資料廣播中的額外資訊（additional information）告知資訊接收端下次資料擷取時間為何時，藉以調整行動客戶端之能源消耗模式，使之一步步讀取廣播資料物件直到最後取得所要之資料為止，達到行動客戶端省電的目的。不過以上之能源問題並非我們主要討論的環節，而我們把研究重心放置在演算法上，討論如何在無線廣播環境下縮短使用者讀取資料的時間，讓使用者能快速的得到其所需求的資訊。



## 第二節 研究目標與限制

資料庫廣播的提出是在資料循環計劃 ( Datacycle project ) [6]中提出，其中談到的通信媒介為高速的光纖架構，而查詢的方法則是運用硬體的裝置進行光纖廣播通道中資料的過濾。而在[17]中已經改進相關資料循環結構，在於廣播頻道之中的資料維護時只使用廣播時相關所需的資料。因著廣播在資訊傳播上的便利性，其形式已經被使用於相關廣播的產品上了[2]。

而在現實生活之中，客戶端所要求的資料種類大小並不一定相等，無線廣播研究中，資料長度較大的資料物件，其廣播所需要的時間必較長。為求研究環境之單純化，而將資料物件的長度皆假設為同一大小，也就是在廣播頻道中的每個單位時間都可以播出一的資料物件或是資料頁，其資料大小皆相同。依照每一位使用者的需求數目來分，當一位使用者下了一個要求給伺服器端，當這個要求只提出一個資料物件的需求，則我們可稱之為「單一要求」( single query ) 的廣播模式；反之如果使用者下的要求超過一個以上的資料物件，則稱之為「多重要求」( complex query ) 的廣播模式。伺服器分析以往備查詢的歷史資料，運用學者研究的排程技術，將這些資料物件作一個有次序的排列，最後運用廣播循環 ( Broadcast Cycle ) 的方式進行循序播放各個資料物件。而該項廣播排程資料物件擺置之研究方向又再細分成為廣播資料物件不重複的「單調式」( flat ) [1]廣播方式與廣播頻道中

廣播資料物件重複播放的「非單調式」( non-flat ) [1]廣播方式。在我們假設前提之下，每個資料物件大小相同，然後參考查詢頻率 ( access frequency ) 依照不同的演算方法做出資料物件排序，資料物件經過排列且每一種資料物件在廣播排程之中並不重複出現，稱之為「單調式」廣播模式；反之，將物件依照出現頻率之多寡或資料元件相互之關係，進行熱門資料物件多重插入廣播排程資料之中，稱之為「非單調式」廣播模式。

早期的研究，以「單一要求」以「單調式」的廣播模式為開端，實驗的環境最簡單，所要考慮的狀況也最少，而後，「單一要求」以「非單調式」的形式廣播模式[3][6]研究較多。不過以現實環境而言，每一個使用者提出需求時，依次要求多個資訊種類是較合乎常理的。因此進而討論「多重要求」的廣播模式，在每個使用者都可能要求多個資訊種類的情況之下，所要求的各種不同的資訊組合情況之中，也有可能存在不同資訊組合中相同資訊物件重複的情況。因此本篇研究重點置於「多重要求」以「單調式」的廣播模式，為求研究環境之單純化，將資料物件的長度皆假設為同一大小，針對每個資訊種類組合，依照資訊種類組合的頻率高低來進行排序，參考各種要求調整的因素進行資料物件位址的擺放，運用相關邏輯演算法則，可以使得平均等待時間最小，滿足資訊使用者的需求。

在我們的廣播資料庫發行方法之中所運用的資料庫型態定義在關

聯式資料庫，而我們將關聯式資料庫中的每一個資料欄位，所包含的資料總和都稱之為一個資料集 ( Dataset )，也可稱它為資料物件 ( Data object )，而我們所討論廣播頻道中的資料廣播順序，指的也就是資料表中資料欄的相關順序。使用者端就是依據廣播伺服器所組成的廣播資料順序進行資料的擷取動作。但是資料的擷取會因著廣播的順序而有差異，如果廣播的資料順序與使用者端的資料擷取順序差異太大，則使用者端的讀取時間則相對較長。因此，為了避免使用者端等待資料的閒置時間過長，不同的廣播資料排列順序會影響到使用者端的平均存取時間，所以較佳的廣播排程資料物件排列順序是我們在無線廣播中尋求的目標。

[14]提到將結構化查詢語言的查詢轉換成為格式化[SA,JA,PA]的查詢模式，SA ( select attributes ) 將where-clause的屬性設定成為  $x = c$ ， $x$  為屬性， $=$  是對照的操作，而 $c$ 則是常數。JA ( join attributes ) 則將where-clause的屬性設定成為  $x = y$ ， $x$ 和 $y$ 為屬性。PA ( project attributes ) 則設為select-clause的屬性。如果SA和JA的交集及PA不是空白的，我們就將SA,JA,PA中重複設定的屬性除去。在查詢條件區塊之中，指出查詢資料的處理順序。即查詢方程式之中SA的屬性較JA的屬性優先被處理，而PA中的屬性責備列為最後處理的項目。而SA,JA或PA中的屬性並不存在任何處理的先後順序性。而被查詢的資料表的相關屬性如下：

資料表 $A=(a_1,a_2)$ ，資料表 $B=(b_1,b_2)$ ，資料表 $C=(c_1,c_2)$

而廣播之中的相關屬性的廣播順序則稱之為廣播程序，例如：  
 $\langle a_1,b_1,b_2,a_2,c_1,c_2 \rangle$ 或是 $\langle a_2,b_2,b_1,c_1,a_1,c_2 \rangle$ 都有可能是廣播程序。我們根據下列查詢方程式來分析：

Select  $a_2,b_2$

From  $A,B$

Where  $a_1 < 20$  and  $a_1 = b_1$

在這個查詢之中， $SA=\{a_1\}$ ， $JA=\{b_1\}$ （因為 $a_1 \in SA$ ，這是從 $JA$ 中區分出來的）， $PA=\{a_2,b_2\}$ 。使用者端首先轉到廣播頻道去下載 $a_1$ 和選擇廣播程序中相關操作屬性 $a_1$ 的值，然後使用者端下載 $b_1$ 和連結操作屬性相關廣播頻道中相關廣播程序中 $a_1$ 的值，然後下載 $b_1$ 和連結操作屬性相關廣播頻道中相關廣播程序中 $a_1$ 的相對應值。之後，我們得到一對資料欄的數目，這個關係著 $A$ 和 $B$ 的資料欄位就連結在一起且滿足在where-clause中的查詢條件。從這些資料欄的數目之中，相關廣播程序屬性 $a_2$ 和 $b_2$ 的資料就被依順序的下載至使用者端，回應了相關使用者的查詢程式。

本篇論文之研究方向，並不探討如何節省接收端的電力需求，抑或是接收端的記憶體限制。本研究目的在使讀取時間（access time）最佳化，進行「多重要求」以「單調式」無資料擷取順序性的廣播模式的演算法研究。所以我們運用有效的演算法，依照資料物件的查詢

頻率，進行廣播資料物件集中化的動作，達到有效的資料循環排程。而模擬實驗中顯示與[4]所提出的演算法有明顯的改善，在均勻分布的使用者查詢模擬環境之中，總存取時間的改善幅度可達 20% ~ 23%；另外在常態分布的使用者查詢模擬環境之中，總存取時間的改善幅度可達 22% ~ 24%。因此我們的研究目標，是希望在於運用較佳的廣播排程，提昇無線廣播使用者端讀取資料的效能，更期望能夠在這方面的研究領域有進一步的突破。

### 第三節 論文架構

本篇第二章進行研究環境之問題描述，詳述演算環境問題之定義以及邏輯演算之環境參數定義。第三章是我們的演算法的介紹，針對相關演算流程進行詳細解釋，如何進行資料的排序、群組、加權及加權排序，最後得到最佳資料物件排序方式。第四章是實驗結果以及實驗效能比較與討論。第五章是進行結論與未來研究方向的探討

## 第二章 問題描述

在廣播資料庫的研究領域之中，我們要解決的問題，是無線廣播頻道中廣播資料的排程領域。因為廣播頻寬的不足，容易引起資料的壅塞，並且容易斷訊，造成通訊品質不佳。為了提昇無線廣播資料物件的查詢效率，所以我們針對先前所介紹之廣播資料查詢環境，進行「多重要求」以「單調式」的特定廣播模式中存取效率之研究，其中我們主要針對無線廣播環境之廣播排程進行改善。

日常生活中各個使用者端進行資料查詢，以及擷取廣播頻道資料的動態，我們可以由圖 2 廣播排程架構圖中，清楚得知整個廣播的資料循環。以圖 2 中所示 *Client 1* 需求的資料物件為  $\{d_1, d_3, d_5\}$ 、*Client 2* 需求的資料物件為  $\{d_2, d_3, d_7\}$  以及 *Client 3* 需求的資料物件為  $\{d_6, d_2, d_5\}$ 。每個使用者端經由上傳頻道，傳送所需要的資訊要求到廣播伺服器，而伺服器將需要進行廣播的資料物件經由邏輯演算後，產生最佳化的廣播資料物件排序  $\langle d_6, d_2, d_7, d_1, d_5, d_3, \dots, d_N \rangle$ ，同時排程處理器依照廣播資料物件的排序，將需要進行廣播的資料物件傳送至廣播頻道上，進行一定時間中，相同廣播資料物件循環的重複廣播，即形成所謂的廣播循環排程。使用者則藉由監聽廣播頻道進行資料接收，直到其所有需求之資訊完整接收完畢為止。

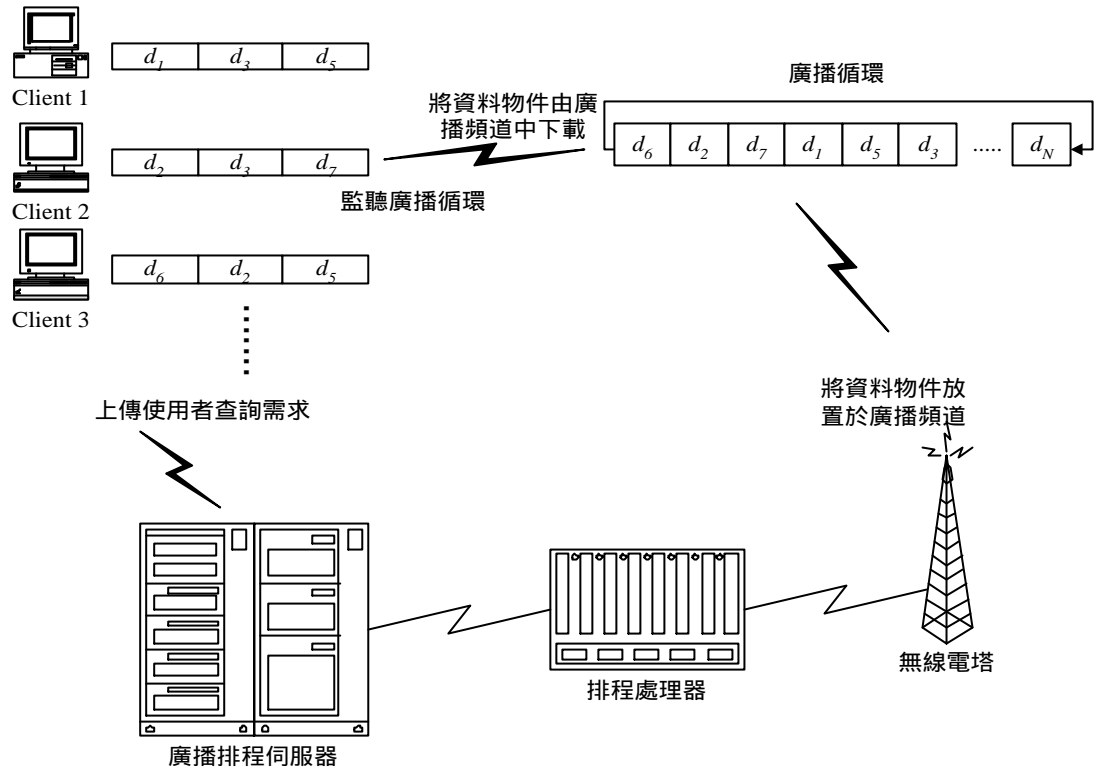


圖 2 廣播架構圖

符號	意義
$d_i$	某一廣播資料物件(含頻率高之重複廣播資料)
$ d_i $	$d_i$ 資料的大小
$D$	所有廣播資料物件的集合 $\{d_1, d_N\}$
$ D $	所有廣播所包含資料物件的數量
$BCL$	廣播一個循環的長度(大小) 例： $\sum  d_i , \forall d_i \in D$
$q_i$	相關廣播資料串流中的單一查詢
$QDS(q_i)$	$q_i$ 的資料物件集
$ QDS(q_i) $	$q_i$ 的資料物件集所包含資料物件的數量
$freq(q_i)$	$q_i$ 的查詢頻率
$Q$	所有查詢的集合 $\{q_1, q_2, \dots, q_m\}$
$\sigma$	某一廣播資料集合 $D$ 的廣播排程

表 1 符號定義



為要進行本篇廣播邏輯演算法之研究，首先我們提出與本研究相關的符號表，如表 1，這些符號將陸續出現在本研究文章內。在資料廣播問題中，我們針對廣播環境探討進行以下假設：

假設：某一使用者所要求的查詢  $q_i$ ，它所包含的資料物件集  $QDS(q_i)$  中的資料物件排序是無次序性的擺置。例如  $QDS(q_i)=\{d_1, d_3, d_6\}$ ，使用者在無線廣播進行資料傳遞時，使用者擷取資料時無須依資料查詢順序進行廣播資料的下載，也就是由廣播排程之中將所需求的資料，依照廣播的排程順序將所有的需求依次下載。

無線廣播的廣播頻道資料排程問題，主要是在探討廣播主機如何產生最佳的廣播排程，使得使用者端的資料總存取時間（*Total Access Time, TAT*）最小化，而 *TAT* 的計算[4]呈現如下：

$$TAT(\mathbf{s}') = \sum_{q_i \in Q} AT^{avg}(q_i, \mathbf{s}') \times freq(q_i) \quad (2-1)$$

其中  $AT^{avg}(q_i, \mathbf{s}')$  表示查詢  $q_i$  在廣播排程  $\sigma'$  之中的平均存取時間。

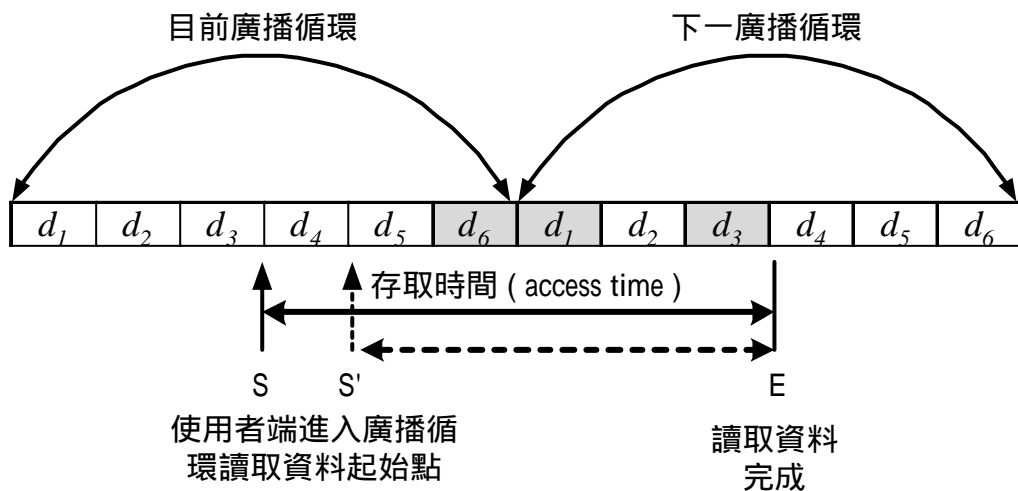


圖 3 單一查詢於無線廣播中的呈現

我們運用圖 3 中較為單純的廣播排程，模擬不同查詢使用者端進入廣播排程中接收廣播資料的狀況，以解釋使用者端進行廣播存取。圖 3 呈現，某查詢  $q_i$  所需資料物件為  $\{d_1, d_3, d_6\}$ ，因著本研究的假設，由廣播排程之中將所需求的資料，依照廣播的排程順序將所有的需求依次下載。其中 S 使用者端與 S' 使用者端，所要查詢的資料物件與查詢  $q_i$  所需資料物件相同，但 S 進入廣播循環的時間，與 S' 進入廣播循環的時間點並不相同。因此從中我們觀察到，因為查詢資料的使用者，進入廣播循環中的讀取資料的時間點並不一致，所以使用者端接收資料物件的平均存取時間 ( $AT^{avg}$ ) 的時間較難量測。因此 *Yon Dohn Chung* 和 *Myoung Ho Kim* 在 [4] 中有定義運用查詢資料間隔距離 (*Query Distance, QD*)，來進行資料物件相對距離的量測。也因為資料間隔距離加大，資料讀取時間相對也增加，由此可見  $QD$  的長短與  $AT^{avg}$  呈現正相關的發展。而我們同時考慮到廣播排程是一個重複的廣播資料循環，所以跨越不同的廣播循環進行資料擷取是非常有可能存在的。

我們假設  $|QDS(q_i)|=n$ ，則  $QDS(q_i)$  的廣播排程資料循環的順序則表示為  $\langle d_{i_1}, d_{i_2}, \dots, d_{i_n} \rangle$ ，其中  $g_k$  則代表廣播排程  $s'$  中資料物件  $d_{i_k}$  到  $d_{i_{k+1}}$  兩者之間的資料間距，因此我們將  $QD$  定義如下：

$$QD(q_i, s') = BCL - \text{MAX}(g_k), \quad 1 \leq k \leq n. \quad (2-2)$$

以圖 3 為例， $QDS(q_i)$  的  $QD$  計算方式如下：

$$s' = \langle d_1, d_2, d_3, d_4, d_5, d_6 \rangle$$

$$QDS(q_i) = \{d_1, d_3, d_6\}$$

$$BCL=6$$

$$g_1 = d_1 \text{ 與 } d_3 \text{ 之間的資料間距} = 1$$

$$g_2 = d_3 \text{ 與 } d_6 \text{ 之間的資料間距} = 2$$

$$g_3 = d_6 \text{ 與 } d_1 \text{ 之間的資料間距} = 0 \rightarrow \text{跨廣播排程}$$

$$\text{則取得 } MAX(g_1, g_2, g_3) = 2$$

$$\text{因此 } QD(q_i, s') = 6 - 2 = 4$$

所以我們得到  $QDS(q_i)$  的  $QD$  單位長度為 4。

因著  $QD$  的長短與  $AT^{avg}$  呈現正相關的發展，且  $QD$  的長短較  $AT^{avg}$  較易計算，所以本篇論文運用  $QD$  的長短，作為實驗比較以及量測的標準，將總查詢資料間隔距離 (*Total Query Distance, TQD*) 的最小化，作為本篇論文努力的目標。本篇論文中的  $TQD$  計算如下所示：

$$TQD(s') = \sum_{q_i \in Q} QD(q_i, s') \times freq(q_i) \quad (2-3)$$

## 第三章 我們的演算法

本篇演算法必須先針對相關使用者的查詢資料，進行相關廣播資料庫中資料物件的前置處理，之後將這些廣播資料物件輸入排程演算法之中，進行廣播排程最佳化的處理，最後即得到最有效之廣播循環。以下我們就藉著相關廣播查詢使用者端的查詢資料，進行本篇演算法的介紹。

### 第一節 初步廣播資料處理

當使用者端將資料傳送至廣播主機時，各個資料物件的讀取次數有一定程度上的差異，因此廣播主機蒐集各個使用者查詢後，將各個被查詢的資料物件，依照其被查詢的讀取次數，進行廣播物件隨機的排列，取得初步的資料廣播順序。

我們以八組  $QDS(q_1) = \{d_2, d_3, d_5\}$ 、 $QDS(q_2) = \{d_4, d_7\}$ 、 $QDS(q_3) = \{d_1, d_3\}$ 、 $QDS(q_4) = \{d_{13}, d_9\}$ 、 $QDS(q_5) = \{d_{11}, d_{13}, d_9\}$ 、 $QDS(q_6) = \{d_{10}, d_{12}\}$ 、 $QDS(q_7) = \{d_6, d_{11}, d_8\}$ 、 $QDS(q_8) = \{d_6, d_8, d_{12}\}$  的使用者查詢資料物件集為例。各組的查詢頻率分別為  $freq(q_1) = 21$ 、 $freq(q_2) = 17$ 、 $freq(q_3) = 8$ 、 $freq(q_4) = 13$ 、 $freq(q_5) = 5$ 、 $freq(q_6) = 33$ 、 $freq(q_7) = 11$ 、 $freq(q_8) = 26$ 。當查詢資料上傳至主機後，我們將各個被查詢之資料物件查詢頻率進行加總紀錄，再將所有被查詢之資料物件進行隨機的廣播順序擺置，如圖 4 所示。

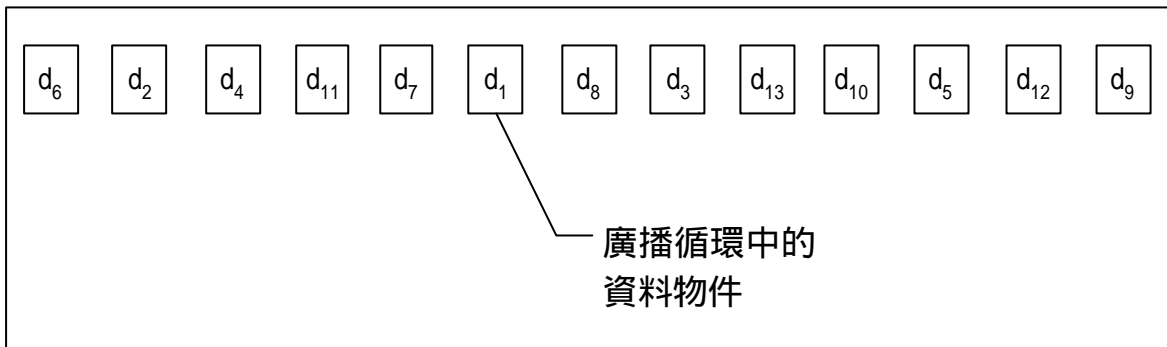


圖 4 初步廣播資料物件順序配置圖

## 第二節 廣播資料子群組化

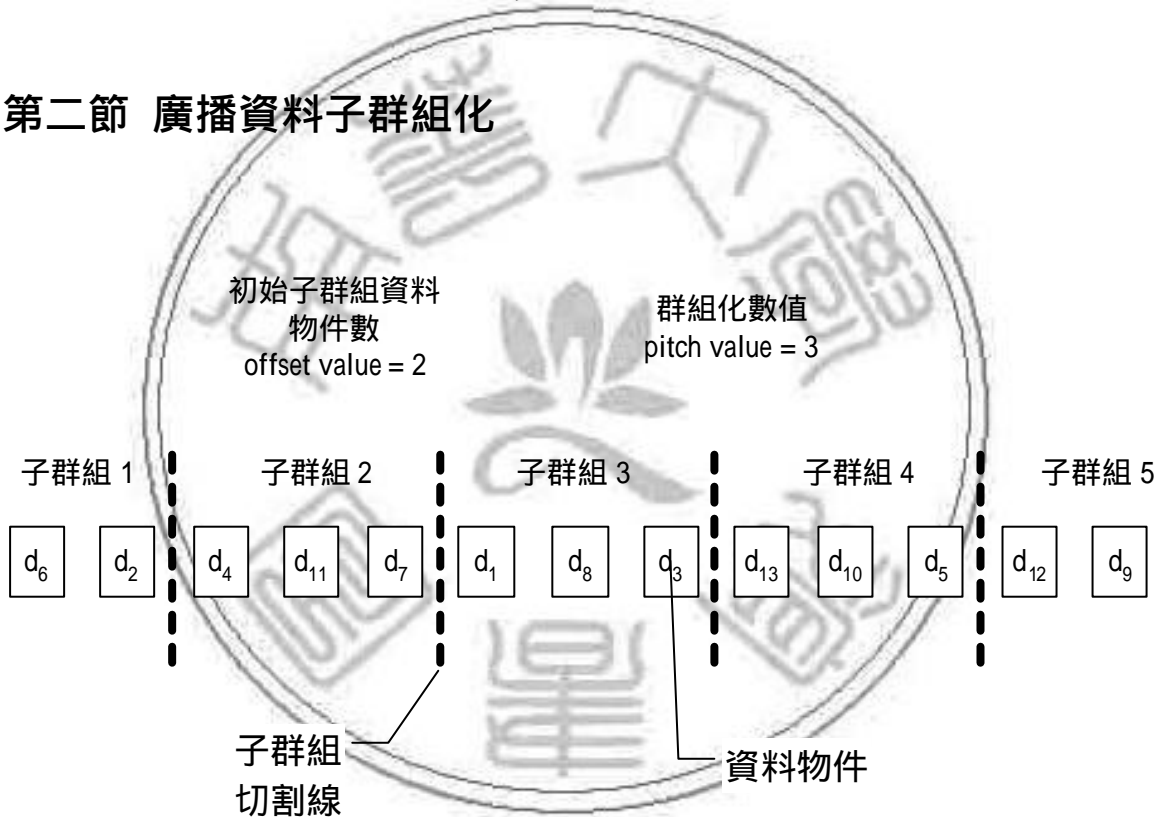


圖 5 子群組切割圖

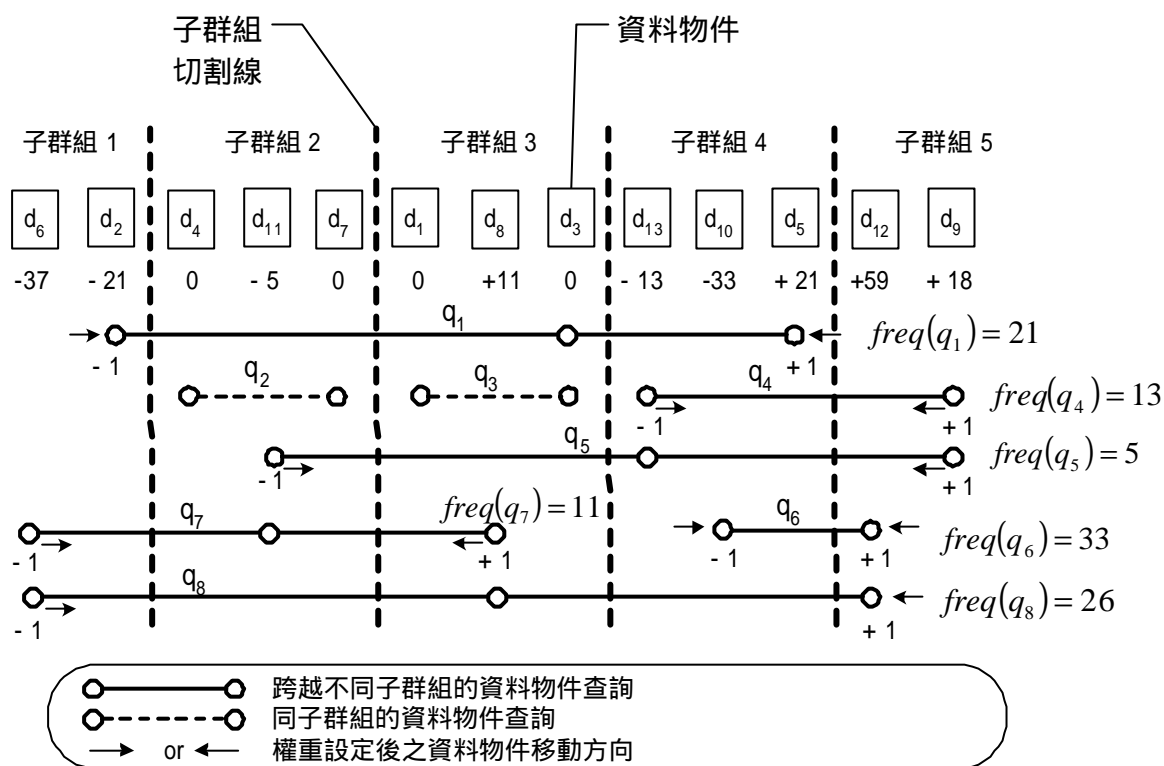
廣播物件經由任意排序之後，下一步則是進行廣播資料子群組化的動作。進行廣播資料子群組化的動作時，是運用群組化數值 (*pitch value, p*) 所產生的切割線，進行區分資料子群組的動作。而廣播資料物件則由起始廣播物件，至廣播排程最後一個廣播資料物件，皆進行廣播

資料群組化的動作。

而第一個的群組其中所包含的資料物件數目，是我們運用隨機取值的方式取界於  $1 \sim p$  之間的數值，產生所需的第一個群組資料物件數 (*offset value, o*)，形成第一個資料子群組。其餘廣播資料物件的群組化過程，則是運用  $p$  值作為各個群組的資料物件數大小，進行資料物件子群組化的切割動作，依序完成廣播排程中，資料物件的子群組化。最後若是資料物件個數不足，則相同視為一個完整的資料物件子群組。

圖 5 呈現群組化數值  $p = 3$ ，而運用隨機取值的方式得到第一個群組資料物件數  $o = 2$ ，因而得到群組內容為 2 個資料物件所組成的第一個子群組。再經由運用  $p$  值切割的動作後，則形成了 3 組由三個資料物件所組成之完整的子群組。最後僅存的 2 個資料物件，則形成了最後的廣播資料子群組。經由上述的群組切割模式，我們總共得到了五組的廣播資料子群組。

### 第三節 廣播資料權重設定



我們從圖 6 呈現相關查詢的分布狀況，因著子群組的切割，我們將跨越不同子群組的使用者查詢，運用實線連接各個被查詢資料物件的方式呈現，而存在於子群組之中的使用者查詢，則運用虛線連接各個被查詢資料物件的方式呈現。存在於子群組之中的使用者查詢，其資料物件分布呈現較為集中，而跨群組的使用者查詢，則呈現較為分散的資料物件分布。在本篇的演算過程中，所要進行的動作是進行使用者查詢資料物件集中化的動作，因此我們運游子群組切割的方式，判斷使用者查詢的資料物件是否集中，再進一步將資料物件較為分散的使用者查詢，進

行權重值 ( *weight value, w* ) 的設定，以提供資料物件集中運算使用。

因為資料物件較為集中的使用者查詢，若進行資料再集中的動作是沒有意義的，因此我們使用虛線呈現該使用者查詢，將之略過資料物件集中的運算邏輯。而需要進行資料物件集中運算的使用者查詢，我們將該使用者查詢的最左端的資料物件，進行負權重值的設定，而最右端的資料物件，則進行正權重值的設定，其餘資料物件的權重值則為 0。

以圖 6 而言，雖然各個查詢資料物件串的權重設定時以+1 或是-1 來呈現相關演算中資料物件權重的值，但是使用者查詢的數量並非單純只有單一使用者進行查詢，因此我們再進行各查詢的權重設定時，其權重的設定必須再乘上該項查詢的使用者查詢頻率，才能忠實反應實際的使用者端查詢訴求。所以實際的權重值  $w$  計算方式如 3-1 所示。

$$w(d_i, q_i) = \begin{cases} +1 \times freq(q_i) \\ -1 \times freq(q_i) \\ 0 \end{cases} \quad (3-1)$$

, where  $q_i \in Q$

就算各個使用者查詢的資料物件，已經各自算出各自查詢的資料物件頻率，最後還必須經過加總各個資料物件權重，才能真正作為後來要進行的排序動作。而累計權重的計算方式如 3-2 所示。

$$w_{d_i} = \sum_{d_i \in QDS(q_j)} w(d_i, q_j) \quad (3-2)$$

接下來我們直接運用表 2 的方式，呈現相關使用者查詢資料物件權重計算後的權重值，最後再進行資料物件的權重累積。



	$d_6$	$d_2$	$d_4$	$d_{11}$	$d_7$	$d_1$	$d_8$	$d_3$	$d_{13}$	$d_{10}$	$d_5$	$d_{12}$	$d_9$
$q_1$	0	-21	0	0	0	0	0	0	0	0	+21	0	0
$q_2$	0	0	0	0	0	0	0	0	0	0	0	0	0
$q_3$	0	0	0	0	0	0	0	0	0	0	0	0	0
$q_4$	0	0	0	0	0	0	0	0	-13	0	0	0	+13
$q_5$	0	0	0	-5	0	0	0	0	0	0	0	0	+5
$q_6$	0	0	0	0	0	0	0	0	0	-33	0	+33	0
$q_7$	-11	0	0	0	0	0	+11	0	0	0	0	0	0
$q_8$	-26	0	0	0	0	0	0	0	0	0	0	+26	0
$w_{d_i}$	-37	-21	0	-5	0	0	+11	0	-13	-33	+21	+59	+18

表 2 資料物件權重計算

圖 6 中呈現 13 個資料物件以及八條查詢資料串，其中有六條查詢資料串跨越不同的廣播子群組。該資料串我們以實線來呈現該資料串是必須進行相關權重計算使用。圖中實線的資料串部分，也只有最左邊的資料物件，以及最右邊的資料物件進行權重計算，因此總共有 12 個資料物件進行權重計算。我們將其中所有的查詢資料串權重值皆單純化為”1”，但資料物件的權重是必須將查詢資料串的權重乘上該查詢的頻率，才會得到真正該查詢中相關資料物件的權重值。

因為資料物件的查詢並不一定單純只存在於單一查詢之中。以圖 6 之中的資料物件  $d_9$  為例，它就呈現在兩個不同的查詢資料串 ( $q_5, q_6$ ) 之中，因此不同的查詢資料串流之中只要資料物件相同，就必須將各個查詢串流中相同資料物件的權重進行加總。所以  $d_9$  的權重值計算就如以下的運算方式為  $w_{d_9} = 1 \times freq(q_4) + 1 \times freq(q_5) = 13 + 5 = 18$ ，而所有計算後之廣播排程資料物件相關權重則如表 2 所呈現。

#### 第四節 廣播子群組內資料排序

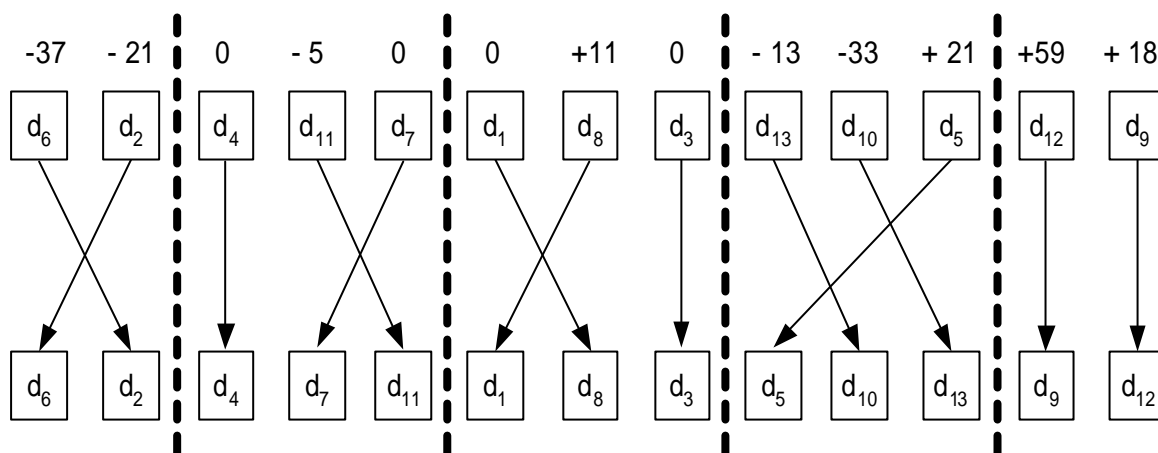


圖 7 權重計算及子群組內資料物件排序法

我們根據廣播排程中資料物件的權重值，進行各個廣播子群組中資料物件的排序。如果該子群組中的資料物件有較大的正權重值，則該資料則向左移動；反之，該權重值為負值時，則將資料物件向右移動。就子群組中的資料物件排序而言，權重則是呈現由左至右的降冪排列。其中我們所進行的降冪排序的資料演算，因著查詢數量可固定在某一定的範圍之中，所以我們可使用 *Counting Sort* 這樣較快速且運算效能較好的排序演算方式，使排序的效能儘可能的逼近  $O(n)$ ，以減少本篇演算法整體資料運算流程所需的時間。

## 第五節 排程演算法及其參數設定

演算過程之中，有相關的參數是可進行變動調整的。其中群組化數值  $p$  是運用於廣播資料物件，進行子群組化切割時使用。因此進行初步廣播資料切割時，本篇會運用總資料物件數乘上運用隨機取值的方式取界於  $0 \sim 1$  之間的數值  $a$  作為  $p$  值，如此一來可較容易切割成為三組子群組，而同時各資料物件的權重設定才比較不會為  $0$ ，也才能進行子群組中的資料排序。其次，本演算法是運用多次循環切割以及排序的動作，以進行資料的集中化。因此，我們會運用固定的循環參數 ( $s$ ) 在該  $p$  值下進行  $s$  次的運算循環，以求得較佳的資料集中效果。

演算過程中我們將群組化數值  $p$  由大到小進行一定比例的轉變，而於單一  $p$  值進行切割同時，我們運用循環參數  $s$  進行  $s$  次的同一  $p$  值中資料子群組化、廣播資料物件權重設定，以及子群組內資料排序以上三者間的重複動作，以取得當次運用  $p$  值排序的最佳解。而相關子群組資料物件一再的進行排序的動作，不只有  $p$  值進行數值的變動，而是於同時時間運用隨機運算，取得第一個群組資料物件數 ( $o$ ) 的方式，進行子群組切割線的挪移，進一步重複子群組資料物件的排序。總結步驟如下：

1. 資料物件子群組化（運用隨機取得  $o$  的方式進行子群組切割）
2. 計算所有廣播資料物件的權重值
3. 廣播子群組中資料物件排序

最後經由  $s$  次的群組擾動後，再進行  $p$  值的縮減，重複上述的步驟。而所有的邏輯演算排程皆因著  $p$  值的改變進行不斷的演算，而廣播資料物件的擾動也隨之漸趨減緩。

因此本篇論文演算法呈現如下：

$D$ : 所有廣播資料的集合

*Set the initial subgroup pitch value  $p = |D| * \mathbf{a}$      $/* 0 < \mathbf{a} < 1 */$*

*Do {*

*$i = 1;$*

*Do {*

*Subgroup Generation (randomly choose offset  $\mathbf{o}$ )*

*Accumulate weight value for each Data Object*

*Sorting within Subgroups*

*$i = i + 1;$*

*} while (  $i \leq s$  )     $/* s$ :每一個固定 *pitch value* 的循環次數參數  $*/$*

*$p = p - ( 0.9 * ( p / \log p ) )$*

*} while (  $p \geq 2.0$  ) (3-3)*

## 第四章 實驗結果與討論

### 第一節 模擬環境

我們的模擬平台，所使用的硬體為 Celeron®、CPU 1.6G、RAM 512MB，所使用的作業系統平台為 Windows2000Server，模擬程式開發使用 VB6.0。

在我們所設定的廣播環境之中，每一個使用者端皆可透過上傳頻道送出要求到伺服器端，而所送出要求的資訊可以是可包含兩個以上的資料物件，而這些資料物件是「固定大小」的，而伺服器端接收到這些「多重需求」後，會放置在伺服器之中進行廣播資料查詢頻率的計算，最後經由固定時間的廣播排程演算，將資料物件運用廣播相關排程演算法進行廣播資料物件排程設定，當可被廣播的資訊決定後，此資訊將被送往廣播的頻道進行廣播，取代現行廣播排程之內容。而使用者端則可以透過監聽廣播頻道，當自己所需要的資料到達時即透過下載頻道接收資料，直到所有需求的資訊都被滿足。我們假設使用者端皆可儲存任何在廣播頻道中所需求的資料，並且所有資料皆可於廣播頻道中獲得，因此如何縮短使用者端的平均資料存取時間，則是我們所要研究的方向。而在一切的假設之中，我們簡化所有的環境，將焦點集中於解決廣播排程所縮短的存取時間，進行計算總存取時間來評估廣播的效能。

## 第二節 實驗資料產生

名稱	設定值	單位
資料庫大小	100,200,300,400,800,1000,1200 1400,1600	種
使用者查詢種類	100,150,200,250,300,350,400,450,500 600,700	個數/要求
查詢選擇率	0.02, 0.03, 0.04, 0.05 0.003,0.004,0.005,0.006,0.007	百分比
循環運算參數( $s$ )	50,100,200,400	次數
起始 $p$ 值設定參數( $a$ )	$0 < a < 1$	

表 3 實驗資料條設定數據

在無線廣播的環境之中，首先我們必須先造出實驗用的資料檔，而這些資料檔最主要是要用來模擬用戶端對於資料物件需求的情形。就查詢頻率分佈狀況而言，實驗環境我們使用了均勻分布 (*uniform*) 以及常態分布 (*normal*) 來模擬現實生活中使用者的查詢狀況。

我們所運用的實驗環境中，使用均勻分布主要是模擬資料主機所廣播的資料物件，是一般使用者都有興趣的資料，譬如：每日頭條新聞或是每日氣象預測等大眾化的資訊提供，因此使用者所查詢的資料物件頻率比較會呈現均勻分布的狀況。再者，若是運用隨機取樣的方式進行廣播資料的查詢分配，其最後所形成的廣播資料物件查詢頻率統計也會趨近於均勻分布的狀態。

其次，我們使用常態分布的查詢資料環境進行實驗，目的在模擬一般廣播環境中的資料查詢頻率，因為資料查詢一定有其熱門資料，相對也存在冷門資料，因此建置常態分布的資料環境，是為了要符合一般廣播使用者查詢的習慣所建置的。

### 第三節 廣播效能比較

經過無線廣播模擬之後，我們運用本文所提出的演算方法所得到的 *TQD*，比較 [4] 所提出的 *Greedy* 演算法與本論文演算法 (*Efficient Algorithm, EA*) 之優劣好壞。以下我們就被廣播的資料庫物件數、使用者查詢種類，以及查詢選擇率來進行廣播效能的比較。

在第一個實驗之中所運用的是均勻分布的使用者查詢資料環境，而我們固定查詢選擇率值為 2%，以及固定 100 筆的使用者查詢種類；進行資料物件數大小 100~400 個資料庫資料物件數的改變。實驗數據得到如下。

Data Objects	Greedy ( TQD )	EA ( TQD )	EA ( TQD reduction)
100	15970	6000	62.43%
200	56280	33070	41.24%
300	104800	73700	29.68%
400	162930	117758	27.73%

表 4 均勻分布中改變查詢資料物件數的實驗數據

實驗結果如圖 8 所示。

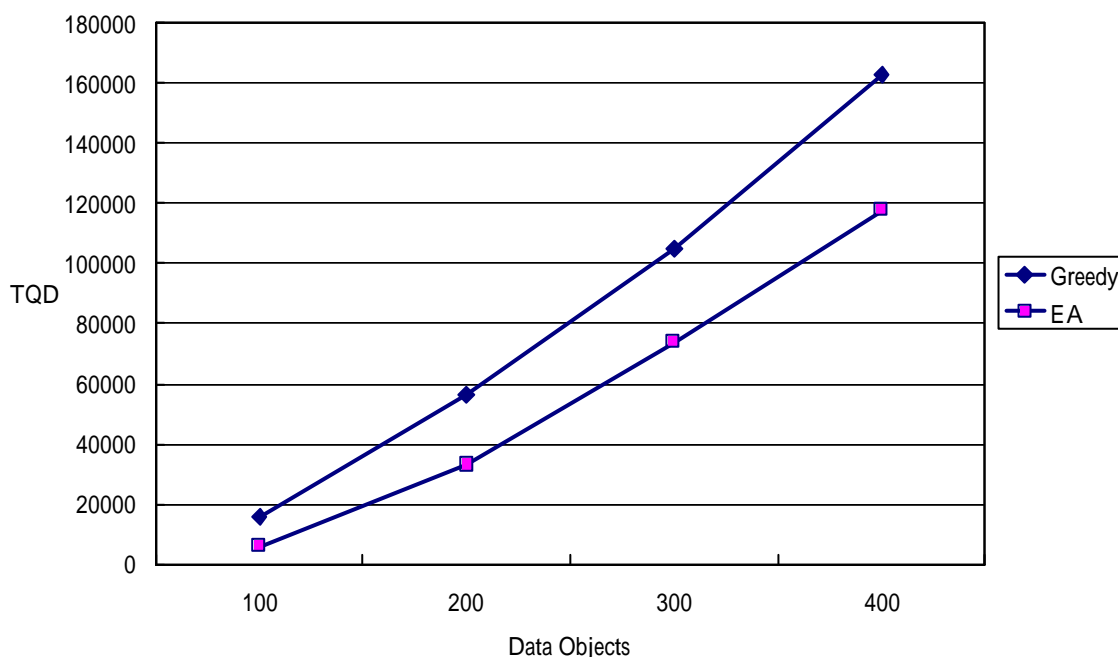


圖 8 均勻分布中改變查詢資料物件數，GREEDY 與 EA 的 TQD 改變情形



如圖 9 所呈現，在本實驗之中，我們固定 100 個使用者查詢種類，但是查詢資料庫資料物件數高時，以及切割迴圈數增加時候，皆呈現較佳的效能提昇，雖然效能的提昇並不顯著，但可以說越多的切割循環與效能的提昇是成正比。因為循環次數的多寡其效能相差並不遠，因此其餘 *EA* 演算法所產生的 *TQD* 資料的比較，皆運用演算後所得的各循環次數 *s* 所得的 *TQD* 平均值進行效能的評比。

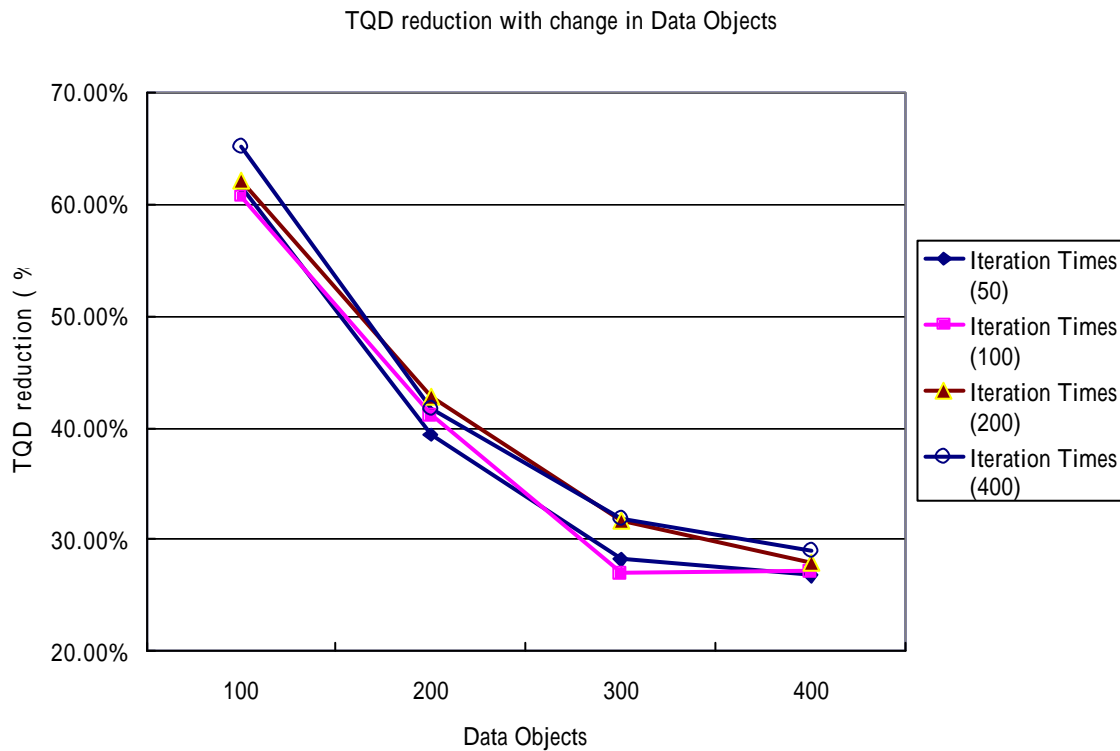


圖 9 均勻分布中每一 *P* 值子群組切割排序循環運算次數 *s* 之變換情形

在第二個實驗之中所運用的是均勻分布的使用者查詢資料環境，而我們固定查詢選擇率值為 2%，以及固定 1000 個資料物件的資料庫大小，但是使用者的查詢數量並不相同。實驗數據得到如下。

Querys	Greedy ( TQD )	EA ( TQD )	EA ( TQD reduction)
100	540760	380310	22.59%
150	937960	703558	24.99%
200	1331850	1067430	19.85%
250	1731260	1427738	17.53%
300	2117300	1832053	13.47%
350	2542650	2204273	13.31%
400	2971640	2600198	12.50%
450	3387090	2988943	11.75%
500	3800520	3392978	10.72%

表 5 均勻分布中改變查詢使用者數量的實驗數據

實驗結果如圖 10 所示。

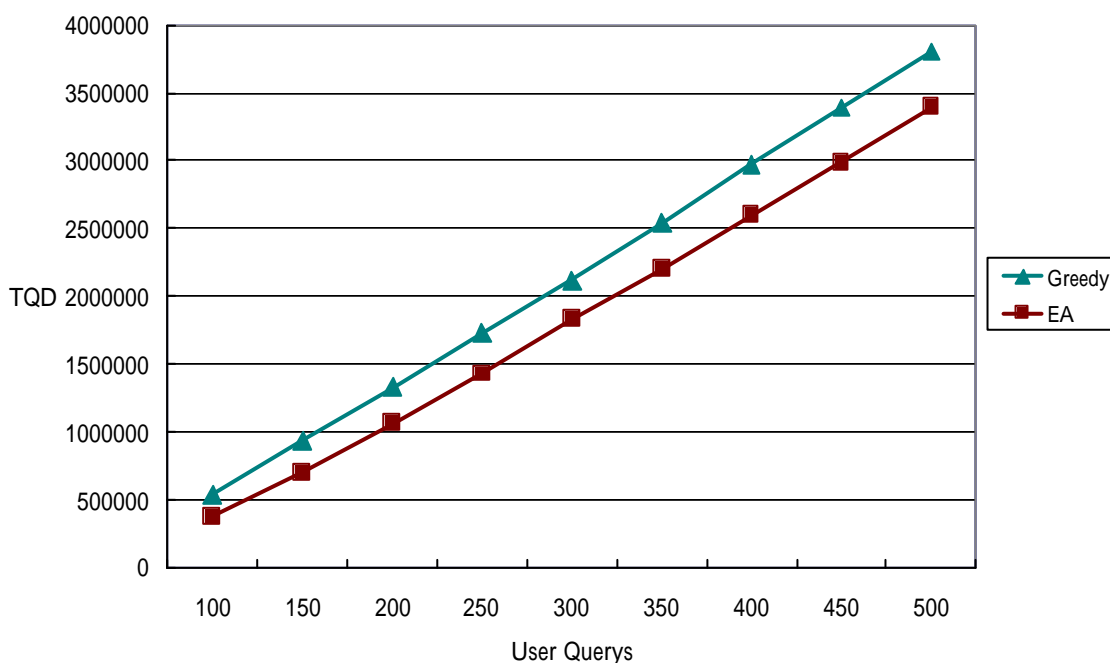


圖 10 均勻分布中改變查詢使用者數量大小，GREEDY 與 EA 的 TQD 改變情形

在第三個實驗之中所運用的同樣是均勻分布的使用者查詢資料環境，而其中我們固定資料庫大小在 1000 個資料物件，以及固定 100 筆的使用者查詢種類；進行查詢選擇率值的改變，所得到的數據如下。

實驗結果如圖 11 所示。

Selectivity (%)	Greedy ( TQD )	EA ( TQD )	EA ( TQD reduction)
2	540760	380568	29.62%
3	659180	510283	22.59%
4	715420	600133	16.11%
5	767860	668518	12.94%

表 6 均勻分布中改變查詢使用者查需選擇率的實驗數據

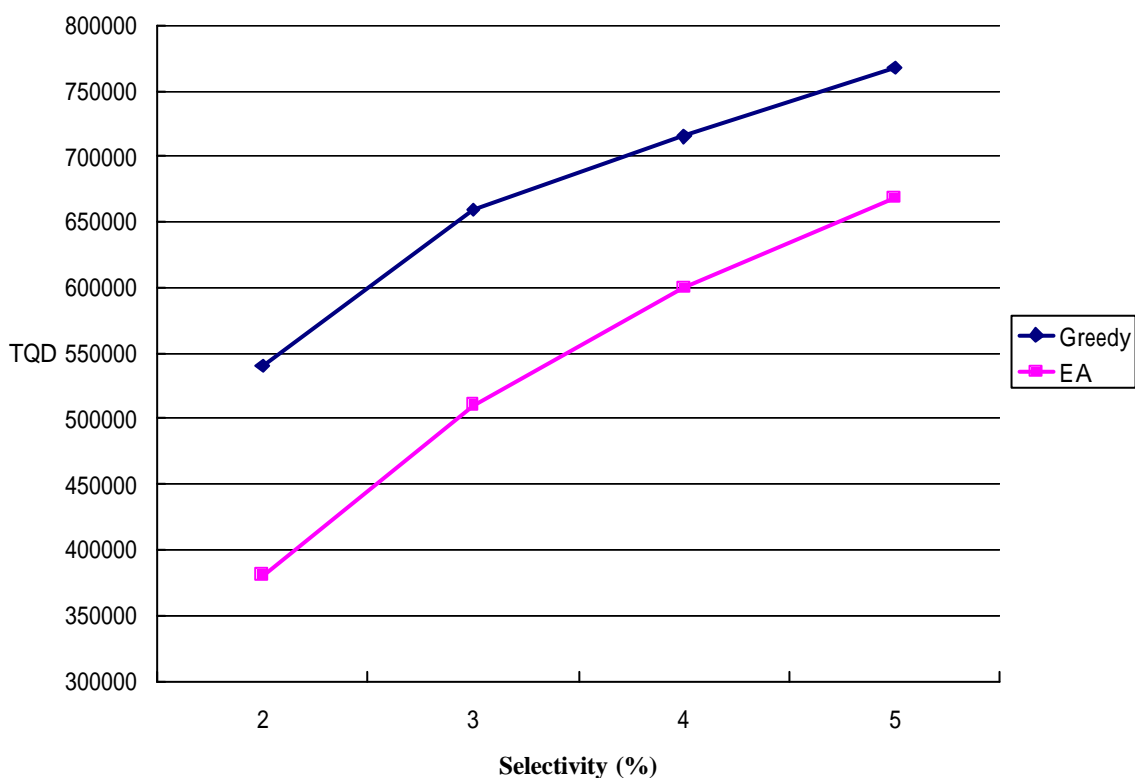


圖 11 均勻分布中改變查詢選擇率，GREEDY 與 EA 的 TQD 改變情形

在第四個實驗之中所運用的是常態分布的使用者查詢資料環境，而我們固定查詢選擇率值為 0.5%，以及固定 500 筆的使用者查詢種類；進行資料物件數大小 800~1600 個資料物件的改變。實驗數據得到如下。

Data Objects	Greedy ( TQD )	EA ( TQD )	EA ( TQD reduction)
800	130893	99323	24.12%
1000	193400	156533	19.06%
1200	255768	212338	16.98%
1400	323965	284943	12.05%
1600	381143	351898	7.67%

表 7 常態分布中改變查詢資料物件數的實驗數據

實驗結果如圖 12 所示。

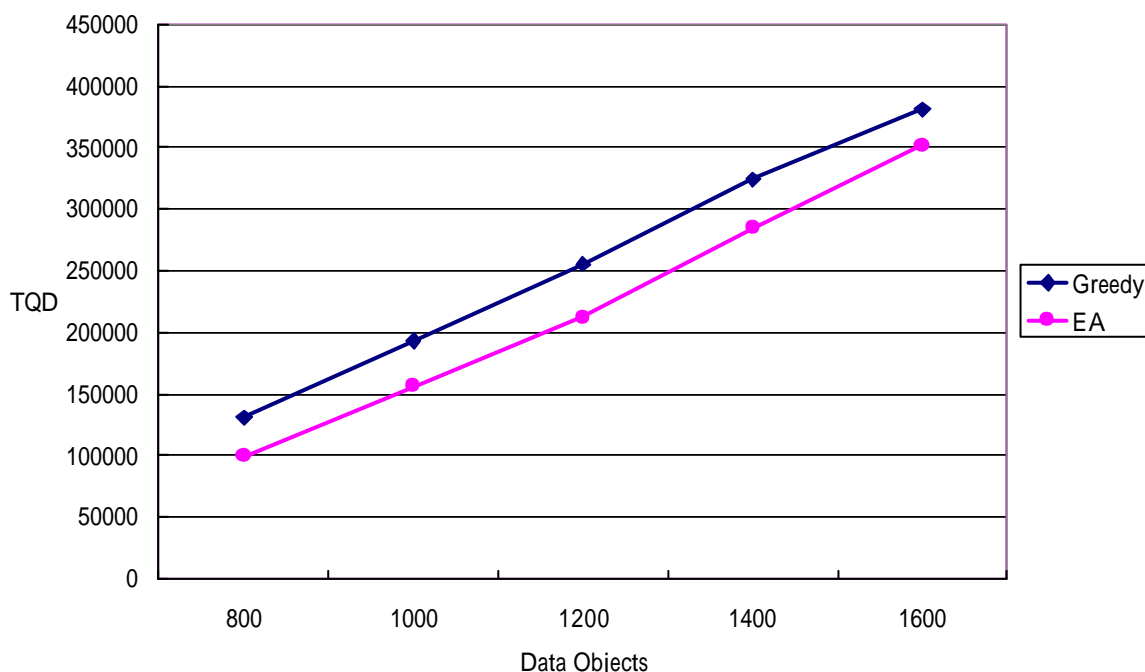


圖 12 常態分布改變查詢資料物件數，GREEDY 與 EA 的 TQD 改變情形

在第五個實驗之中所運用的是常態分布的使用者查詢資料環境，而我們固定查詢選擇率值為 0.5%，以及固定 1000 個資料物件的資料庫大小，但是使用者的查詢數量並不相同。實驗數據得到如下。

Querys	Greedy ( TQD )	EA ( TQD )	EA ( TQD reduction)
300	83197	50321	39.52%
400	137694	104214	24.31%
500	193400	156533	19.06%
600	241924	199459	17.55%
700	300243	245698	18.17%

表 6 常態分布改變查詢使用者數量的實驗數據

實驗結果如圖 13 所示。

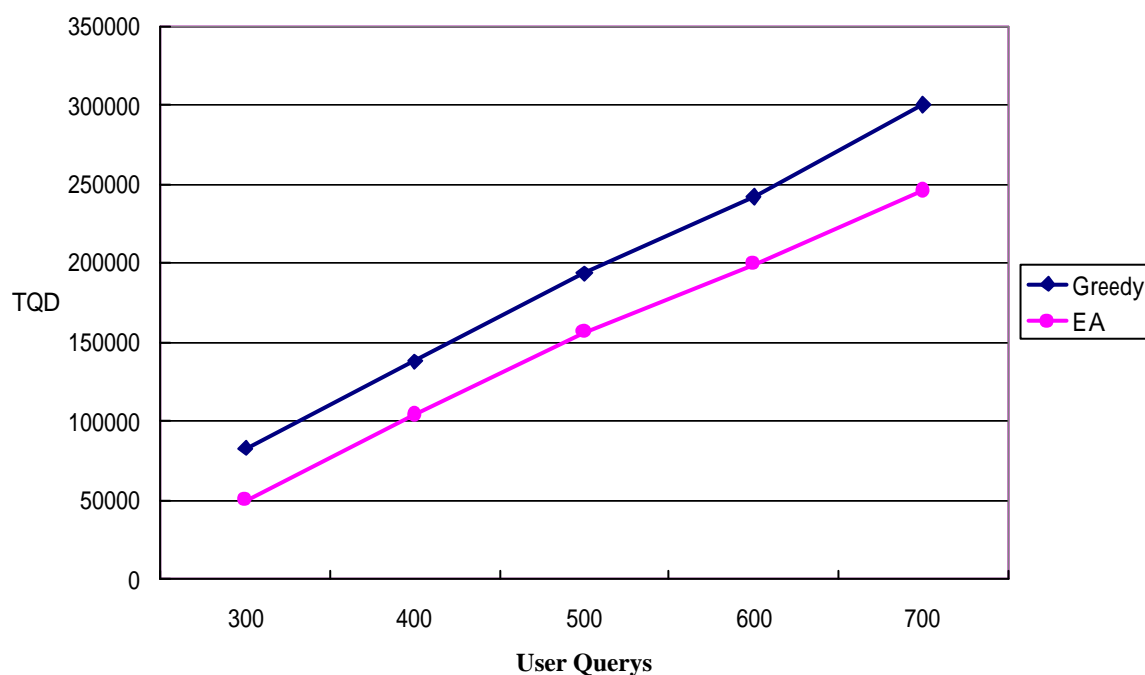


圖 13 常態分布改變查詢使用者數量大小，GREEDY 與 EA 的 TQD 改變情形

在第六個實驗之中所運用的同樣是常態分布的使用者查詢資料環境，而其中我們固定資料庫大小在 1000 個資料物件，以及固定 500 筆的使用者查詢種類；進行查詢選擇率值的改變，所得到的數據如下。實驗結果如圖 14 所示。

Selectivity (%)	Greedy ( TQD )	EA ( TQD )	EA ( TQD reduction)
0.3	84555	29961	64.57%
0.4	144914	100432	30.70%
0.5	193400	156533	19.06%
0.6	226746	197312	12.98%
0.7	257491	233182	9.44%

表 8 常態分布改變查詢使用者查詢選擇率的實驗數據

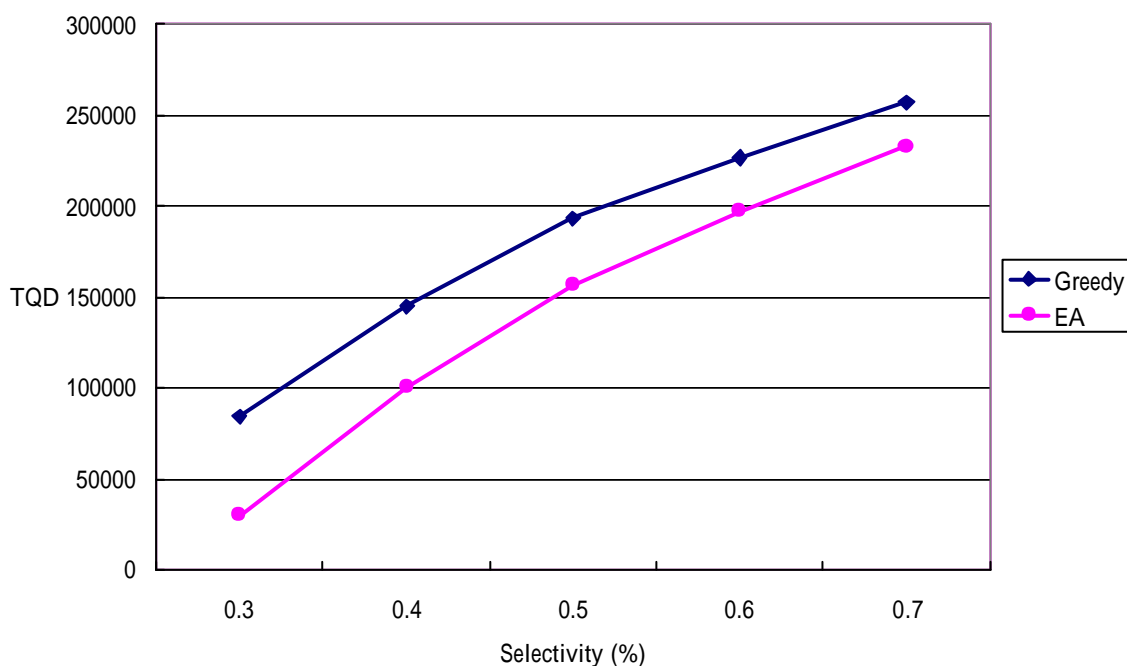


圖 14 常態分布改變查詢選擇率，GREEDY 與 EA 的 TQD 改變情形

最後我們運用均勻分布的使用者查詢資料環境，進行循環參數(s)設定所得到的研究發現。以表 9 以及圖 15 至圖 20 為例，我們從圖 16 至圖 18 中資料物件數的增加，發現為了得到較佳的 *TQD*，其中循環參數(s)的設定，需要隨之增加循環的次數；另從圖 19、20 之中發現使用者查詢選擇率的數值變大，循環次數亦需隨之增加，才能得到較佳的 *TQD*。在各個實驗之中，我們由 50 次的循環值開始之後，一直實驗至 400 次的循環次數，不論增加查詢資料物件數或是增加使用者查詢選擇率，大約在 200 次到 300 次的循環值設定的時候，其 *TQD* 的長度會變化漸趨穩定。因著循環次數的設定越大，本篇演算法的演算時間則越長。所以本篇建議循環值設定於 250 次左右的循環次數即可。既可節省演算法的演算時間，也可得到較佳且較穩定的 *TQD* 值。

Iteration Times		50(s)	100(s)	150(s)	200(s)	250(s)	300(s)	350(s)	400(s)
Data Object	Selectivity (%)	EA ( TQD )							
100	2%	6240	5940	5700	5740	5700	5740	5740	5780
200	2%	34420	33640	32660	32380	31560	32880	32030	32170
300	2%	76170	74630	75170	73430	73120	73600	73770	73200
400	2%	119160	118160	119620	118000	117120	115660	117990	115730
500	2%	160580	160010	160450	160250	159610	159500	159580	159630
100	3%	19400	19360	19480	19390	18900	18520	18530	19420
100	4%	28580	28380	28530	28420	28350	28170	28270	28130
100	5%	39600	39610	39650	39390	39330	39120	39240	39450
100	6%	44410	44030	43810	43140	43140	43320	42960	43180
1000	2%	380310	380740	380570	380100	379980	380090	380010	380120
1000	3%	522120	506500	503640	505060	502200	506250	503100	503450
1000	4%	610750	600220	595200	593430	592560	594740	593570	596130
1000	5%	684130	664390	662620	660370	658700	660590	659150	658180

表 9 相關循環參數設定的實驗數據

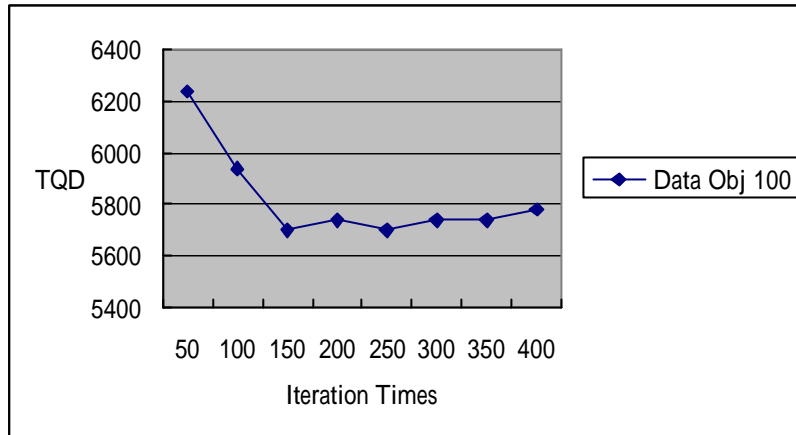


圖 15 循環參數設定改變， $TQD$  的改變情形

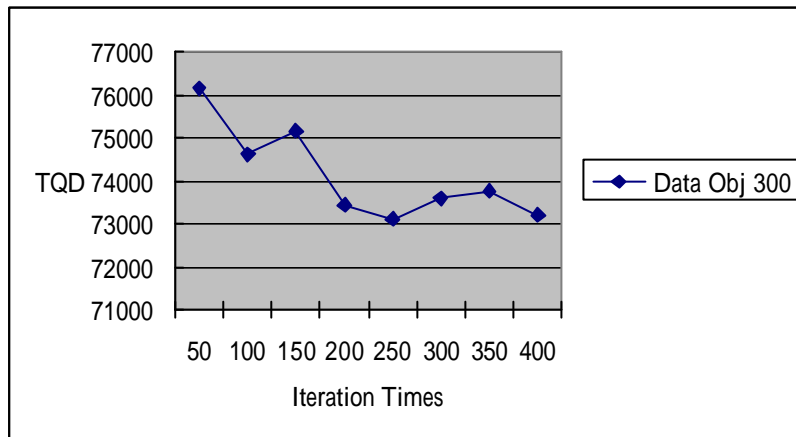


圖 16 循環參數設定改變且資料庫物件數變為 300 時， $TQD$  的改變情形

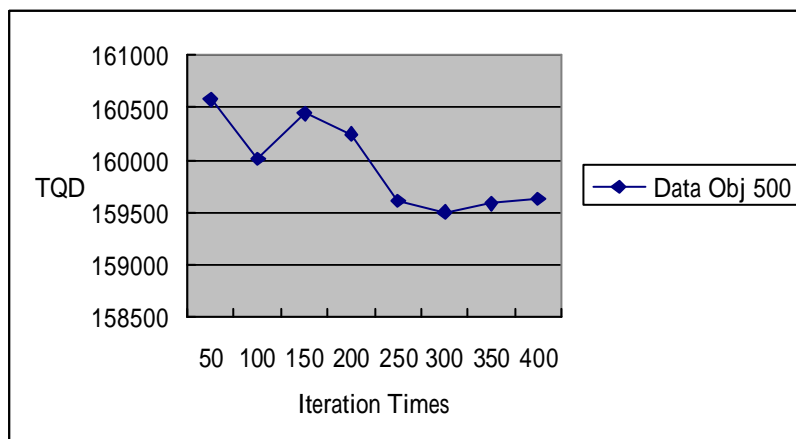


圖 17 循環參數設定改變且資料庫物件數變為 500 時， $TQD$  的改變情形



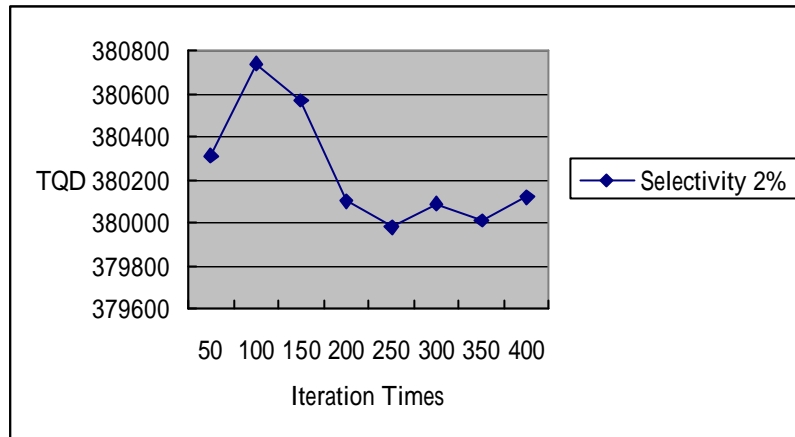


圖 18 循環參數設定改變且資料庫物件數變為 1000 時，*TQD* 的改變情形

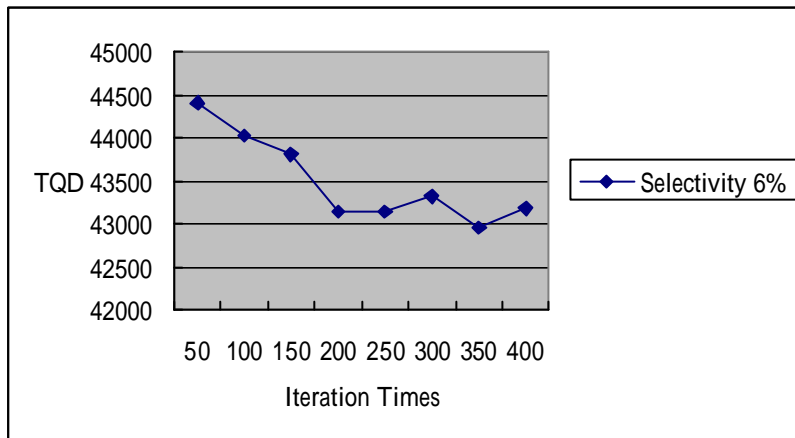


圖 19 循環參數設定改變且使用者查詢選擇率變大時，*TQD* 的改變情形

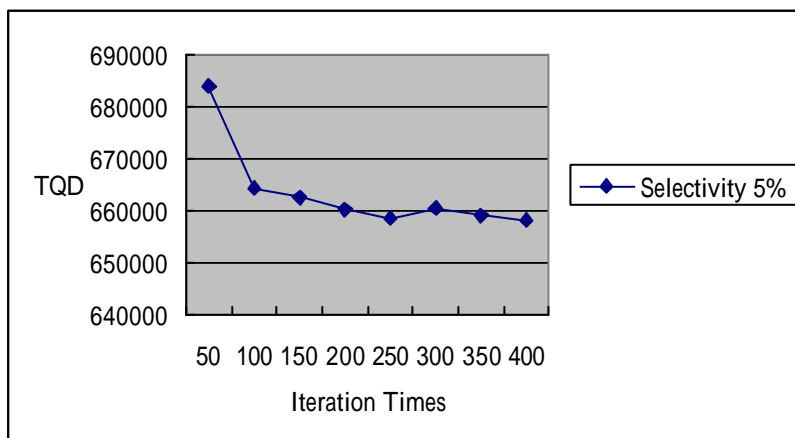


圖 20 循環參數設定改變資料庫物件數變為 1000，且使用者查詢選擇率變大時，*TQD* 的改變情形

就上述的各個實驗中，我們針對均勻分布的查詢環境，得到本篇論文所使用的演算方法較之[4]所提出的演算方法，確實提昇使用者的存取效能約 20% ~ 23%，使得資料得以較快速的讓使用者端進行資料的接收所需之資訊。另就常態分布的查詢環境而言，在相對資料庫的資料物件數越高，或是使用者查詢選擇率越高時，本篇論文所提供的演算方法改善空間並不明顯。所以，針對較單純的使用者端查詢環境而言，本篇所使用的演算法可得到較佳的廣播排程效能，而實驗結果確實證實提昇使用者的存取效能約 22.35% ~ 24.47%。



## 第五章 結論與未來研究方向

為了節省頻寬以及應付大量客戶端同時上線，廣播為最常用的方法。為了縮短客戶端的等待時間，目前主要研究皆是運用適當的演算法，進行廣播資料物件排程的最佳化，使得使用者端可藉著排序好的廣播資料序，較快的得到所需的資料。本文運用演算法針對「多重要求」以「單調式」的廣播模式，進行廣播排程效能最佳化的改善，幫助使用者端迅速的得到所需求的資料，讓 *TAT(Total Access Time)*總存取時間為最短且最快速。

而本篇所使用的演算法比較上，較符合資料查詢頻率群聚的效益，且同時也顧及頻率較低的群組，相關程度上的資料依賴性。本篇演算法在廣播問題中應用時也很切合實際，因為廣播中使用者的需求型態本身就有其群聚查詢頻率，與本篇使用之演算法特性相近，如此運用本篇之演算法，在無線廣播環境之中來解決問題是在適合不過了。

我們所提出的研究範疇，是以「多重要求」以「單調式」的廣播模式為研究環境，考慮的是單一廣播頻道的資料播送，以及單純的廣播資料物件大小，於現實生活中的廣播資料物件大小並不吻合。因此在未來的研究方向，將考慮廣播的資料物件允許不同大小，並且多重廣播頻道環境的廣播模式進行研究，使廣播排程能真正做到符合現實環境生活中使用者端的查詢需求。

## 參 考 文 獻

- [1] S. Acharya, R. Alonso, M. Franklin, and S. Zdonik, "Broadcast Disks: Data Management for Asymmetric Communication Environments," Proc. ACM SIGMOD Conf., pp. 199-210, May 1995.
- [2] D. Aksoy and M.J. Franklin, "Scheduling for Large-Scale On-Demand Data Broadcasting," Proc. IEEE INFOCOM Conf., pp. 651-659, 1998.
- [3] T.F. Bowen, G. Gopal, G. Herman, T. Hickey, K.C. Lee, W.H. Mansfield, J. Raitz, and A. Weinrib. "The datacycle architecture," Communications of the ACM , 35(12) , December 1992
- [4] Yon Dohn Chung , Myoung Ho Kim "Effective Data Placement for Wireless Broadcast" Division of Computer Science Department of Electrical Engineering and Computer Science , Korea Advanced Institute of Science and Technology , 373-1 , Kusung-dong , Yuseong-gu , Taejeon , 305-701 , Korea.
- [5] V. Gondhalekar, Scheduling Periodic Wireless Data Broadcast, M.S.Thesis, The University of Texas at Austin, U.S.A., December, 1995.
- [6] G. Herman, G. Gopal, K.C. Lee, and A. Weinrib, "The datacycle architecture for very high throughput database systems," In Proceedings of the 1987 ACM-SIGMOD International Conference on Management of Data, pages 97-103, June
- [7] C. H. Hsu, G. Lee, and A.L.P. Chen, "A Near Optimal Algorithm for Generating Broadcast Programs on Multiple Channels," Proc. ACM 10th Int'l Conf. Information and Knowledge Management, pp. 303-309, 2001.
- [8] C. H. Hsu, G. Lee, and A.L.P. Chen, "Index and Data Allocation on Multiple Broadcast Channels Considering Data Access Frequencies," Proc. Int'l Conf. Mobile Data Management, pp. 87-93, 2002.
- [9] Chih-Lin Hu, Ming-Syan Chen "Adaptive Balanced Hybrid Data Delivery for Multi-Channel Data Broadcast," IEEE International Conference on Communications, Pages:960 - 964 vol.2, 2002.
- [10] T. Imielinski, S. Viswanathan, and B.R. Badrinath, "Energy Efficient Indexing on Air," Proc. ACM SIGMOD Conf., pp. 25-36, May 1994.
- [11] W.C. Lee and D.L. Lee, "Using Signature Techniques for Information Filtering in Wireless and Mobile Environments," Distributed and Parallel Databases, vol. 4, no. 3, pp. 205-227, July 1996.
- [12] S.C. Lo and A.L.P. Chen, "Optimal Index and Data Allocation in Multiple Broadcast Channels," Proc. 16th IEEE Int'l Conf. Data Eng., pp. 293-302, Feb. 2000.
- [13] S.C. Lo and A.L.P. Chen, "An Adaptive Access Method for Broadcast Data under an Error-Prone Mobile Environment," IEEE Trans. Knowledge and Data Eng., vol. 12, no. 4, pp. 609-620, July/Aug. 2000.

- [14] Guanling Lee, S.C. Lo and A.L.P Chen, "Data Allocation on the Wireless Broadcast Channel for Efficient Query Processing," IEEE Transactions On Computers Special Section on Data Management Systems and Mobiles Computing, October, 2002, volume 51, pp. 1237~1252.
- [15] K.L. Tan and J.X.Yu. "A dynamic scheduler for the infinite air-cache," Data and Knowledge Engineering, 24(1):97-112,1997.
- [16] N. Vaidya and S. Hameed, "Scheduling Data Broadcast in Asymmetric Communication Environments," ACM/Baltzer Wireless Networks, vol. 5, no. 3, pp. 171-182, 1999.
- [17] K.H. Yeung and T.S. Yum, "Selective Broadcast Data Distribution Systems," IEEE Trans. Computers, vol. 46, no. 1, pp. 100-104, Jan. 1997.