

國立交通大學

資訊管理研究所

碩士論文

設計與實作一個基於語義關鍵字搜尋的

點對點網路



The Design and Implementation of a Peer-to-Peer
Network Based on Semantic Keyword Search

研究生：張家安

指導教授：羅濟群博士

中華民國九十四年六月

設計與實作一個基於語義關鍵字搜尋的
點對點網路

The Design and Implementation of a Peer-to-Peer
Network Based on Semantic Keyword Search

研究生：張家安
指導教授：羅濟群

Student: CHANG, CHIA-AN
Advisor: Chi-Chun Lo



A Thesis

Submitted to Institute of Information Management
College of Management

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of

Master of Business Administration

in

Information Management

June 2005

Hsinchu, Taiwan, the Republic of China

中華民國 九十四年六月

設計與實作一個基於語義關鍵字搜尋的 點對點網路

研究生：張家安

指導教授：羅濟群 老師

國立交通大學資訊管理研究所

摘要

點對點網路(Peer-to-Peer Network, 簡稱P2P Network)顧名思義就是一種節點與節點之間可以直接溝通的網路技術。點對點網路不需要集中式的中央控管機制即可讓系統間直接進行資訊和服務的分​​享,這些資源和服務包括了檔案交換(File Sharing)、即時訊息(Instant Message)、分散式運算(Distributed Computing)等等。目前在點對點網路架構之應用上,檔案分享系統為最熱門的應用之一。

本論文提出一種語義關鍵字搜尋的點對點網路架構。語義關鍵字搜尋在集中式架構中,由於可以將語義資料庫放在中央伺服器,因此可以輕鬆的完成;而在點對點網路中,由於節點之間並沒有一個中央伺服器可以存取,因此本論文主要的目的就是要結合超級點網路(Super Peer Network)及傳統集中式架構中所提到之檢​​索技巧—「查詢擴展」(Query Expansion)以期讓語義關鍵字搜尋在點對點網路中能夠實行,且效果能趨近於集中式架構之效果。本論文所提出之架構,若應用在音樂檔案分享系統上。經由模擬實驗證明,確實可讓語義關鍵字搜尋在點對點網路中實行,其效能亦可趨近於集中式架構之效果,而達到系統中訊息傳遞具有方向性、改善語義關鍵字資料庫(Keyword Relationship Database, KRDB)同步的問題。

關鍵字：點對點網路、超級點網路、語義網、關鍵字擷取、資訊擷取

The Design and Implementation of a Peer-to-Peer Network Based on Semantic Keyword Search

Student: CHANG,CHIA-AN

Advisor: Chi-Chun Lo

Institute of Information Management
College of Management
National Chiao Tung University



Abstract

Peer-to-Peer (P2P) network architecture proposes a new technology to exchange information and service between computers without centralized controllers. These resources and service including : files sharing, instant messaging, and distributed computing . Now, P2P file sharing is one of the P2P hottest applications.

This paper proposes a P2P network based on semantic keyword search. It is easy to implement semantic keyword search in “Centralized Network Architecture” because we can put the semantic knowledge base on the centralized server ; however, there is no peer always in the system. Therefore, our main concern is to integrate the advantage of “Super Peer Network” and the “Query Expansion” (an information retrieval method in centralized server-to-client architecture) .Our approach can solve the problems mentioned before. According to the simulation, our approach can provide good performance of semantic keyword searching in P2P system and solve KRDB synchronization efficacious.

Keyword : Peer-to-Peer 、 Super-Node, Semantic Network , Keyword Extraction , Information Retrieval

誌謝

隨著這份論文的完成，也代表著我的學生生涯即將告一段落了。在交大的這兩年，的確讓我改變了不少，不管是做人處事、或者是對事物的態度，我的視野也變得更加遼闊。很感謝這一路上所有幫助過我的人們，真的很感謝你們。

要感謝的人真的很多，首先當然要先感謝我的家人，爸B、媽咪、還有樂樂那麼的支持我，使我能夠順利完成碩士論文的研究；再來就是感謝我的指導教授羅濟群老師，羅老師常常能夠激發我更多的創意，也給了我相當多的建議，在我論文寫作的過程中，如果沒有羅老師的諄諄教誨，我的論文架構將不可能思考得如此周密。再來要感謝的是我的同班同學以及通訊網路實驗室的所有同仁，俊傑學長、立群、一濤、淑雯、道馨、之璿、宗儒等人，很感謝你們，在我有困難需要幫助的時候，總是會伸出援手、互相扶持。最後要感謝的是在我寫論文的時候，一直陪著我的室友以及好朋友，小菁公主還有米腸，謝謝你們一直聽我碎碎唸，還有對我的照顧。

我要感謝的人實在太多了，在此無法一一提及。一份論文的完成，代表著個人的成長，這篇論文能夠通過及發表，除了本身的努力，我的指導教授羅濟群老師還有那些曾經幫助過我的人尤其功不可沒，謝謝你們！

目錄

第一章 緒論.....	1
1.1 研究背景.....	1
1.2 研究動機與目的.....	2
1.3 研究方法.....	3
1.4 論文架構.....	4
第二章 文獻探討.....	5
2.1 點對點網路(Peer-to-Peer Network).....	5
2.1.1 點對點網路之興起與發展沿革.....	5
2.1.2 點對點網路架構的分類.....	6
2.1.3 點對點網路之一般議題以及所面臨之挑戰與困難.....	7
2.2 超級點網路(Super Node Network).....	9
2.2.1 超節點網路.....	9
2.2.2 JXTA.....	9
2.2.3 Skype.....	11
2.2.4 PASS.....	12
2.3 語義關鍵字搜尋(Semantic Keyword Search).....	14
2.3.1 語義關鍵字搜尋.....	14
2.3.2 關鍵字關係資料庫(Keyword Relationship Database).....	14
2.3.2.1 關鍵字關係資料庫的形成.....	15
2.3.2.2 關鍵字關係資料庫之評估與回饋機制.....	16
2.3.2.2 關鍵字關係資料庫之同步機制.....	17
第三章 以 Super Peer 為基礎之語義關鍵字搜尋的點對點網路.....	19
3.1 需求定義.....	19
3.2 系統架構.....	20
3.2.1 分群及 Super Peer 選定準則.....	22
3.2.2 以 Super Peer 為基礎的關鍵字關係資料庫.....	24
3.2.2.1 KRDB 更新之機制.....	24
3.2.2.2 KRDB 同步之機制.....	25
3.2.2.3 搜尋結果評分排行之機制.....	29

3.2.2.4 查詢過的關鍵字之快取機制.....	29
3.3 系統運作流程.....	31
3.3.1 新節點加入系統.....	31
3.3.2 已加入系統之節點更新其分享檔案.....	33
3.3.3 節點送出查詢訊息、查詢命中、搜尋結果排名.....	33
3.3.4 評估回饋及記錄.....	35
3.3.5 KRDB 同步.....	35
3.3.6 節點離開系統.....	36
第四章 模擬實作與成果分析-以音樂檔案分享為例.....	39
4.1 系統環境.....	39
4.2 模擬實作.....	39
4.2.1 系統設計.....	39
4.2.2 模擬假設.....	40
4.2.3 模擬實作之架構.....	40
4.2.4 模擬實作之類別說明.....	41
4.3 系統畫面.....	43
4.4 成果分析.....	45
4.5 與以往方法之比較.....	49
第五章 結論及未來研究方向.....	51
5.1 結論.....	51
5.2 未來研究方向.....	52
附錄.....	53
參考文獻.....	56

圖目次

圖 1 研究方法.....	3
圖 2 簡化、高階視界的點對點網路架構與集中式的主從式網路架構[9].....	6
圖 3 Gnutella網路架構示意圖[2].....	6
圖 4 Super Peer 網路架構[7].....	9
圖 5 JXTA軟體架構[5].....	10
圖 6 JXTA利用Super Peer做訊息轉送流程圖[5].....	11
圖 7 Skype網路架構圖[16].....	12
圖 8 PASS網路架構示意圖[15].....	13
圖 9 點對點語義關鍵字查詢網路架構[11].....	15
圖 10 KRDB產生之過程[11].....	16
圖 11 KRDB評估回饋之過程[11].....	17
圖 12 KRDB同步的流程圖[11].....	17
圖 13 以Super Peer為基礎之語義關鍵字搜尋的點對點網路架構圖.....	20
圖 14 Super Peer Network架構.....	26
圖 15 KRDB同步流程.....	28
圖 16 系統運作流程圖.....	31
圖 17 新的節點加入系統.....	32
圖 18 已加入之節點欲更新其分享之檔案.....	33
圖 19 節點送出查詢.....	34
圖 20 評估回饋及記錄.....	35
圖 21 KRDB同步.....	36
圖 22 分群內一般節點離開系統.....	37
圖 23 分群內Super Peer離開系統.....	37
圖 24 分群內Super Peer與候選點均離開系統.....	38

圖 25 模擬實作之架構圖.....	40
圖 26 模擬程式功能樹.....	41
圖 27 查詢畫面.....	43
圖 28 管理畫面.....	44
圖 29 實驗之metadata正確率.....	48
圖 30 搜尋結果之範例.....	54



表目次

表 1 SP-KRDB之資料結構.....	24
表 2 節點的資料結構.....	27
表 3 查詢記錄表之資料結構.....	29
表 4 搜尋結果實驗之數據.....	47
表 5 SP-KRDB架構與KRDB架構比較表.....	49



第一章 緒論

點對點網路架構(Peer-to-Peer Network, 簡稱以下P2P)是目前一種相當熱門的網路架構。顧名思義,就是一種節點與節點之間可以直接溝通的網路技術。點對點網路不需要集中式的中央控管機制即可讓系統間直接進行資訊和服務分享[1],這些資源和服務包括了檔案交換(File Sharing)、即時訊息(Instant Message)、分散式運算(Distributed Computing)等等。目前在點對點網路架構之應用上,檔案分享系統為最熱門的應用之一。

本論文提出一種語義關鍵字搜尋的點對點網路架構。語義關鍵字搜尋在集中式架構中,由於可以將語義資料庫放在中央伺服器,因此可以輕鬆的完成;而點對點網路為完全分散式的架構,沒有一個集中的地方紀錄及分享這些有用的語義知識,因此使得點與點之間分享有用的語義知識是相當困難的。因此本論文主要的目的是要使得語義關鍵字搜尋在點對點網路中能夠實行且效果能趨於集中式架構之效果。

本章首先探討點對點網路的發展過程中,所面臨到的一些問題與本論文主要的研究目的。



1.1 研究背景

(一) 點對點網路架構

點對點網路架構是目前一種熱門的網路架構,其為網路世界帶來了革命性的影響。點對點網路架構的優點在於資源運用最大化、直接動作,以及資源分享的潛力。點對點網路架構並非近幾年才出現的一項新興技術,其實早期的網際網路基本上就是一種點對點的網路架構模式。只是隨著網際網路的發展,越來越多的網路使用量,使得以伺服器為依據的系統慢慢地變成網路架構的主流。

點對點網路架構簡單來說,就是指在網路上任意的兩個節點可以直接互相連而不必倚靠中央伺服器之通訊技術,最重要的優點是能夠減少中央伺服器的花費,並避免單點伺服器故障所造成的網路問題。在點對點網路架構下,每一節點

(Peer)同時都是一個客戶端(Client)以及一個伺服器(Server)，這代表了在同一時間每一個節點都可以提供服務以及接受服務。

近年來，市場上對於點對點網路的興趣逐漸被提升了。早在 1999 年由美國 Napster 事件所帶動的檔案分享熱潮，已引起相當多的網路研究學者的注意。最近一些熱門應用，如：即時通訊軟體 (Instant Messaging) MSN Messenger、語音交換軟體 Skype、以及檔案分享軟體 Emule、EDonkey 等等點對點軟體的普及，亦可察覺其應用將會帶給人類更多不同的驚喜。

(二) 語義關鍵字搜尋 (Semantic Keyword Search)

關鍵字搜尋是指當使用者在使用檔案分享系統時，為了要搜尋特定資料項目，而輸入關鍵字的動作。由於每個人的認知不同而對於相同的資料項目給予不同的命名，而造成有些目標檔案搜尋不著的情況。語義關鍵字搜尋 (Semantic Keyword Search) 則是指在檔案搜尋過程中，將所有的關鍵字建立起一個關鍵字關係資料庫，當發生上述之情形發生時，則可以順利解決找不到檔案或者是提供檔案但無法分享之情形。

語義關鍵字搜尋在集中式架構中，由於可以將語義資料庫放在中央伺服器，因此可以輕鬆的完成；而在點對點網路中，沒有節點之間並沒有一個中央伺服器可以存取。目前一般市面上的點對點檔案分享軟體，也僅提供單純的「文字匹配搜尋」 (Text-Match Search)。

2004 年 6 月東京大學 Kiyohide Nakauchi 等人提出了點對點語義關鍵字搜尋的論文[11]，其概念為將系統內每一節點所分享之檔案擷取出關鍵字關係列表成為一個關鍵字關係資料庫 (Keyword Relationship Database, 簡稱 KRDB)，藉由此資料庫，將語義關鍵字的概念實作在點對點網路架構上。

1.2 研究動機與目的

在 1.1 節中所提到的 KRDB 架構，其實尚存有許多問題 (請參閱 3.1)，如：KRDB 無法真正同步等。本論文將要針對這些問題提出一個「以 Super Peer 為基礎之語義關鍵字搜尋的點對點網路架構」以解決 KRDB 架構所面臨到的問題。

1.3 研究方法

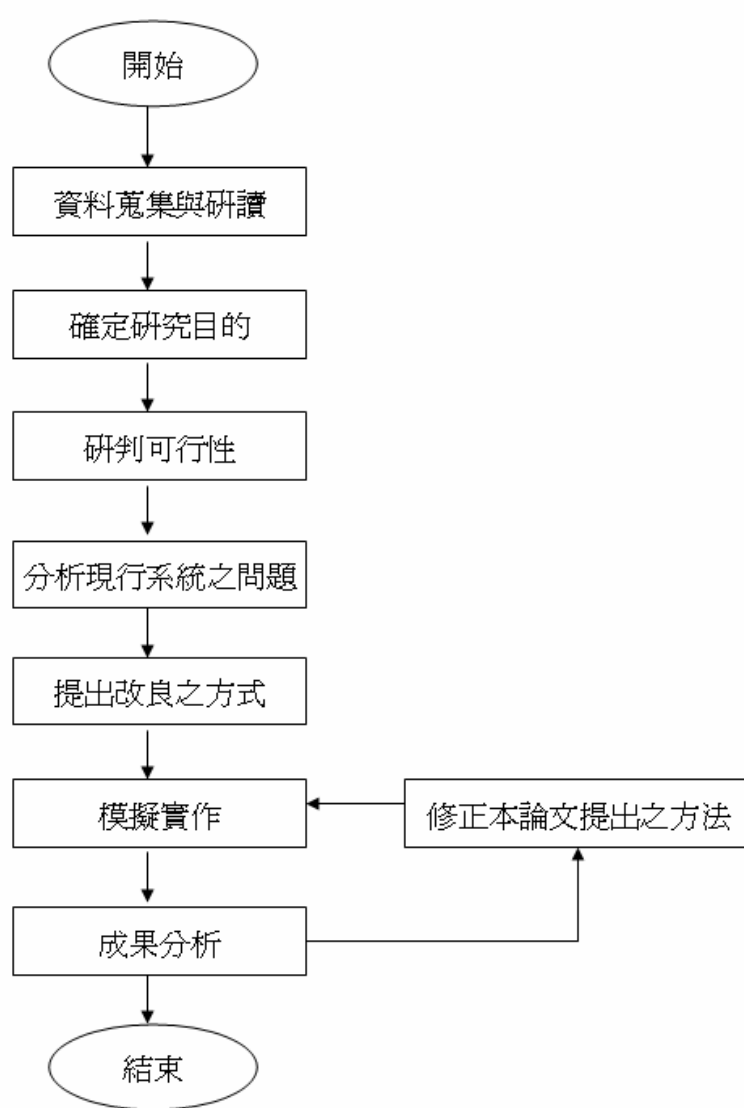


圖 1 研究方法

圖 1 為本論文研究之方法，包括圖中所有步驟。

1.4 論文架構

本論文共分為五章。

- 第一章：緒論
主要介紹本論文之研究背景、研究動機與目的、研究方法以及本論文的架構。
- 第二章：文獻探討
探討本論文相關的文獻。
- 第三章：以Super Peer為基礎的語義關鍵字搜尋點對點網路架構
提出本論文以Super Peer為基礎的點對點語義關鍵字搜尋架構，以及說明系統運作流程。
- 第四章：模擬實作與成果與分析-以音樂檔案分享為例
利用本研究所建立的架構，模擬實作及成果分析；說明本論文系統架構的假設，及對於本研究所開發的架構分別進行成果展現分析其效能，以及可能的研究限制與模擬限制。
- 第五章：結論與未來展望
提出本論文的結論以及未來相關的研究方向。



第二章 文獻探討

現行之點對點網路架構尚有許多的問題。本章將會從點對點資訊分享的現況發展談起，討論其未來發展的方向；接著探討一些在本論文中所需要用到的理論基礎與技術，包括：點對點網路架構(Peer-to-Peer Network)、超級點網路(Super Node或稱Super Peer)、語義關鍵字搜尋 (Semantic Keyword Search)、以及點對點關鍵字資料庫(KRDB: Keyword Relationship Database)。

2.1 點對點網路(Peer-to-Peer Network)

2.1.1 點對點網路之興起與發展沿革

點對點(Peer-to-Peer, 以下簡稱 P2P)網路架構[14]並非一項新的技術，而是一項新的應用技術模式。點對點網路架構是指系統和應用程式都是以完全分散式的方式存在於網路中。由於在這種網路模式下，並不需要一個集中式的控管機制來管理系統資源，而是由系統中所有的點，扮演相同地位的角色，互相溝通及分享資源，因此可以大幅節省購買昂貴的伺服器之花費。而這幾年，逐漸興起的點對點檔案分享軟體以及即時通訊軟體也都是點對點的熱門應用。

點對點網路架構是繼主從式架構(Client-Server)後新興的一種網路應用模式。傳統的主從式架構(圖 2)，系統會有一個或者是以叢集的方式組成的伺服器管理著許多客戶端；而點對點網路則是由系統中所有節點(Peer)所形成。點對點網路架構與主從式架構最大的分別在於節點的角色定位，在點對點網路架構中，所有節點的角色都是對等的，而這些節點互相溝通，彼此互為對方的伺服器或是客戶端。這是點對點網路架構最大的優點，系統並不會因為某特定節點發生錯誤而使得網路停擺；相反的，主從式架構中，系統內有一個中央控管之伺服器，管理著許多客戶端節點，萬一伺服器發生錯誤，則會造成系統無法運作。

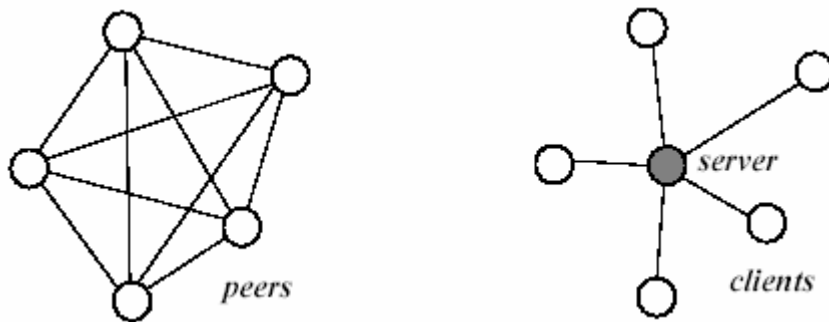


圖 2 簡化、高階視界的點對點網路架構與集中式的主從式網路架構[9]

2.1.2 點對點網路架構的分類

目前點對點網路架構，可分為以下兩種：非結構式點對點網路架構 (Unstructured P2P Network) 以及結構式點對點網路架構 (Structured P2P Network) [14]。非結構式點對點網路架構為目前市面上大部分點對點檔案分享軟體所採用之方向，如：edonkey、emule 等等。非結構式點對點網路採用的搜尋目標檔案之方式為盲目洪流 (Blind Flooding)，因此搜尋速度可能較慢；而後者則是未來研究及邁進的方向，如 CAN[14]、Chord[13]、Pastry[17]、Tapestry[18]、Kademlia[19]、P-Grid[20] 等等，此網路架構利用 Overlay 的觀念，讓搜尋變得更容易。

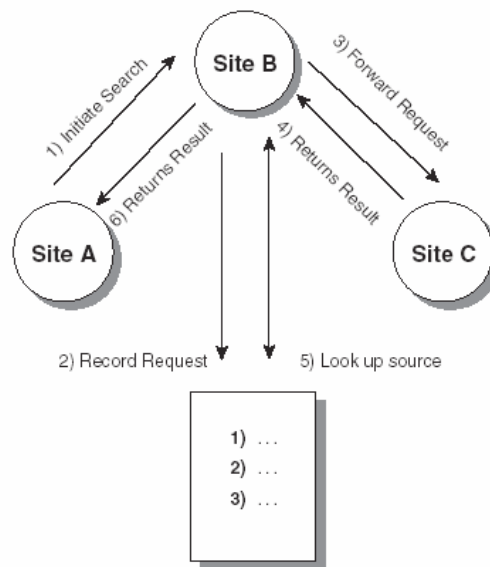


圖 3 Gnutella 網路架構示意圖[2]

非結構式點對點網路架構，較著名之應用有 Gnutella[2]、KaZaA[3]、Napster[6]、以及 Freenet[10]等。Napster 是「客戶端-伺服器架構」，也就是節點會向特定的檔案伺服器詢問以取得目標檔案所在位置。Gnutella[2]與 Freenet[10]則是以廣播方式，意即對所有系統節點發出詢問訊息以達到尋找檔案的目的。然而，這兩種檔案搜尋方式都會使得系統的延展性變差，造成搜尋效率會隨著系統節點數增加而急速下降。

結構式點對點網路架構中，所有節點被組織成為一個邏輯性的網路系統。檔案位置的搜尋則依據邏輯性架構的設計，利用不同的演算法來進行。由於檔案搜尋不需透過特定檔案伺服器或廣播方式，故結構式點對點檔案系統擁有較佳的延展性。此外，由於每個節點因為邏輯架構而擁有其他一些節點的相關資訊，因此結構式點對點檔案系統在檔案搜尋速度以及熱門檔案查詢之負載平衡方面，也有較佳的效能表現。

2.1.3 點對點網路之一般議題以及所面臨之挑戰與困難

點對點網路架構在這幾年日益興盛，而最重要的應用目前來看，不外乎就是點對點檔案分享系統，而這些系統普遍存在著一個問題，就是版權問題[22]。在1999年5月美國波士頓東北大學學生費寧撰寫了一個可讓網路使用者自由交換下載彼此電腦中儲存音樂檔案之點對點軟體，掀起了線上音樂下載的風潮，這個軟體隨後成立了名為Napster的網路公司，而Napster的會員數在成立首年內即超過4,000萬人，其網站流量平均每日可達50萬人次。由於Napster的興起，使得音樂內容的提供者卻無法從中取得如權利金等報酬，同時，線上音樂交換排擠了實體唱片的銷售，因此唱片業者將P2P線上音樂交換視為盜版行徑，RIAA於1999年12月提起對Napster的訴訟，迫使Napster於2001年7月終止所有網路上的歌曲流通交換。近年來，在台灣也有相似的商業模式，像是Ezpeer，以及Kuro等等。不過也都面臨相當嚴重的版權問題。

點對點網路架構在分散式運算的應用上也日漸受到重視。點對點網路架構，主要就是希望能夠集合世界各地每台電腦那些沒被利用到的運算資源，作妥善的利用，利用網路結合個人電腦中間置的運算能力。SETI@home[4](Search for

Extraterrestrial Intelligence at home)正是一個很成功的點對點網路架構的應用的例子。此計畫為美國加州大學柏克萊分校的David Gedye 及 Craig Kasnoff 於1996年7月所提出的計畫。本計畫主要目的是為了求證地球之外是否還有其他的智慧生命存在，利用由波多黎各的Arecibo天文無線電望遠鏡，這是世界上最大和最敏感的射電望遠鏡，而這些由望遠鏡傳回來的訊號需要經過龐大的運算，本計畫結合眾多分散於世界各地的電腦，分析有無不正常之訊息。經過這幾年的努力，推出可以使眾多人來參與的客戶端程式，此程式即是利用P2P的網路架構來運作的。近年來點對點為基礎之分散式運算越來越興盛，就像最近的愛因斯坦的重力波計算也是繼SETI@home後一個成功的例子。但是在此成功的背後仍有許多問題需要尚待解決，例如：如何確保其資料傳送過程中沒有被竊改等等。

另外，點對點網路的搜尋效果可能無法和主從式架構相比，這是由於語義關鍵字搜尋在集中式架構中，由於可以將語義資料庫放在中央伺服器，因此可以輕鬆的完成；而在點對點網路中，沒有節點之間並沒有一個中央伺服器可以存取，關於此點，2-3節將會有更完整的說明。



2.2 超級點網路(Super Node Network)

2.2.1 超節點網路

在超級節點網路[7] [8] [9]中會有一個管理分群內所有節點資訊的節點，此責任較重的節點稱為超節點(Super Node或Super Peer)，如圖4所示，此架構結合了主從式架構的優點(如：搜尋效率)、以及分散式搜尋的優點(如：系統能夠達到負載平衡的狀態及避免單一伺服器受到攻擊)。如圖4所示，圖4(a)中各分群內只有一個Super Node；圖4(b)中各分群內則有兩個Super Peer之超級點網路架構。

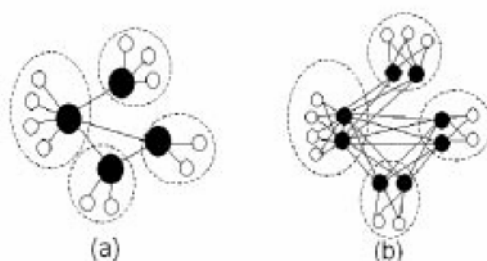


圖 4 Super Peer 網路架構[7]

2.2.2 JXTA

JXTA[15]是昇陽(Sun Corp.)公司提出的一套開放原始碼版本的點對點網路協定，其定義了一個通用且為三層的點對點網路之軟體架構，可讓同一網路上的任何裝置(手機、PDA、PC、伺服器)進行資料的交換與聯繫。JXTA是Juxtapose(同步之意)一詞的簡寫，這個名稱意味著點對點網路架構與主從式網路架構都具有同等地位。

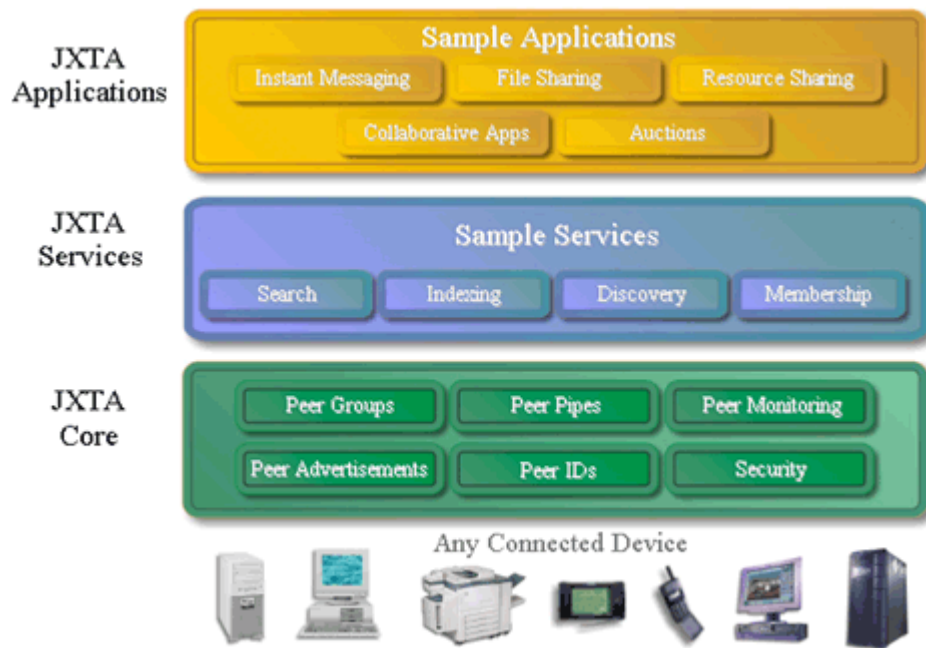


圖 5 JXTA 軟體架構[5]

如圖5，JXTA擁有了六種基於XML(XML-based)的協定，包括：節點發現協定(Peer Discovery Protocol)、節點解答協定(Peer Resolver Protocol)、節點資訊協定(Peer Information Protocol)、節點會員協定(Peer Membership Protocol)、節點黏合協定(Peer Binding Protocol)、節點路由協定(Endpoint Routing Protocol)、以及一些抽象性的概念，例如：節點群組協定(Peer Groups)、管線(pipes)、公佈(advertisement)機制以提供一個單一應用平台給所有使用點對點技術之使用者。

JXTA計畫是昇陽在2001年開始提出的研究計畫，此計畫期望透過點對點網路架構探索分散式網路運算未來的遠景。JXTA架構中利用了Super Peer進行訊息轉送，圖6為其機制之示意圖，JXTA架構中將所有節點分為數個分群，而各分群內有Super Peer作為其領導者，此Super Peer可將收到之訊息轉送給其他Super Peer。

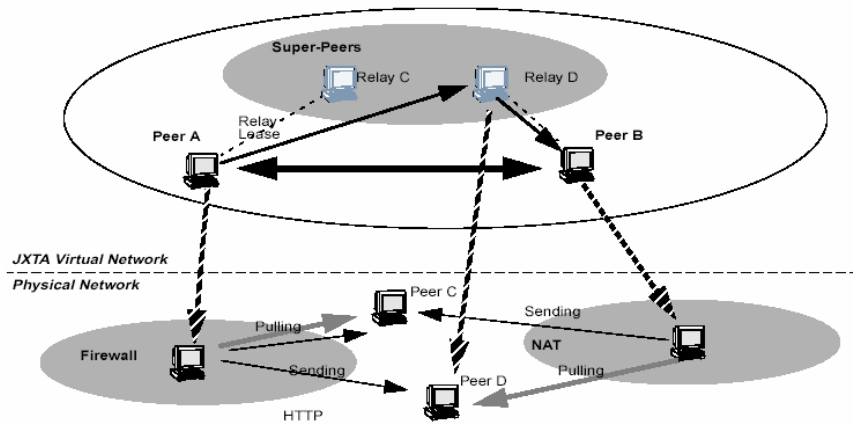


圖 6 JXTA 利用 Super Peer 做訊息轉送流程圖[5]

2.2.3 Skype

Skype 是一個點對點網路架構的語音交換軟體[24]。目前市面上即時通訊軟體均支援語音對話的功能，但 Skype 相對於其他即時通訊軟體的優勢在於相容於現有的防火牆、路由器、NAT 裝置，而且無須對這些網路裝置作任何設定。這意味著不論是否有防火牆或 NAT 裝置的網路環境，Skype 都能夠正常地執行以及使用語音對話。

圖7為Skype的網路架構圖，其架構中也有所謂的Super Node，以作為其他節點之跳板或者是越過防火牆之用途。

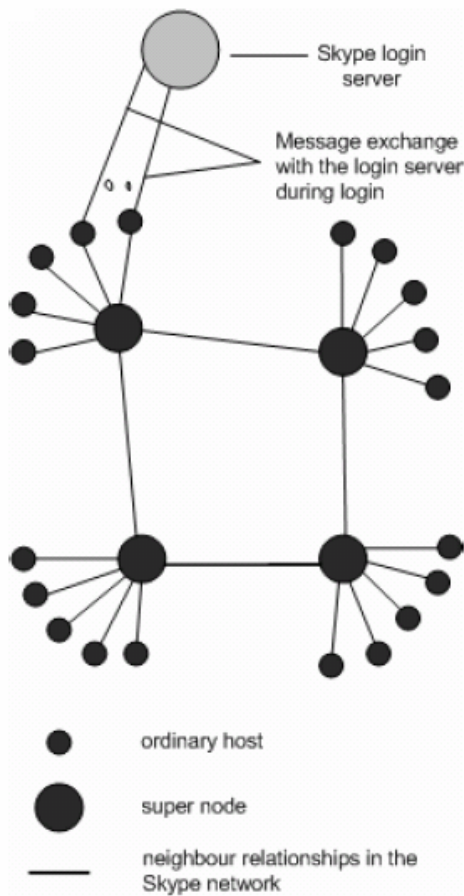


圖 7 Skype 網路架構圖[16]

Skype與以往MSN等IM(Instant Message)工具最大的不同在於除了使用者登入之外，其餘的工作均不依賴中央服務器，Skype在穿透防火牆通訊時完全使用了點對點的架構。圖7中的小黑點是客戶端，大黑點則為超級節點(Super Node)（用途為提供客戶端作為登入踏板及廣播服務），灰色的節點則為Skype的登入服務器。

2.2.4 PASS

PASS[15]為一套點對點檔案分享系統，如圖8，其架構考慮到網路的階層性質，發展了一套透過選出超節點(Super Peer)負責組織子網路與訊息繞送，不僅減低了通訊與計算的資訊，更增進了區域性的性質，且網路可以自我組織，此指當網路成型之時，可保證在數個跳躍數就找到目標節點。但是它的壞處是超節點使用了大量的記憶體空間記錄使用端資料，並需要「快取」(Cache)之前搜尋過的回傳結果。另外，當超節點尚未完全建立完善時，會退化成氾洪法(flooding)

的廣播型式，使得搜尋較無效率的運作方式。

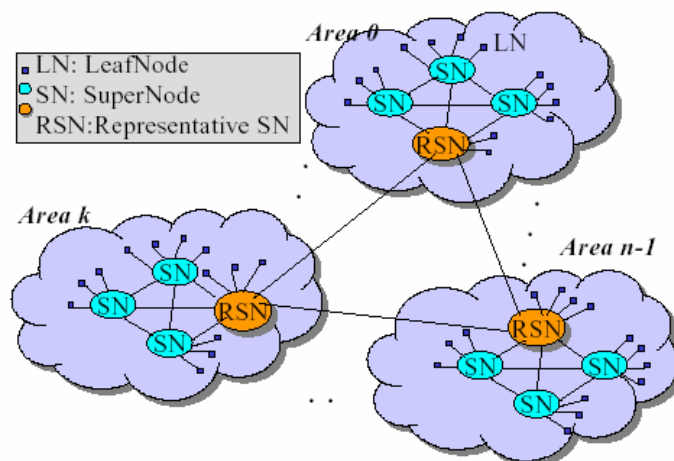


圖 8 PASS 網路架構示意圖[15]



2.3 語義關鍵字搜尋(Semantic Keyword Search)

2.3.1 語義關鍵字搜尋

關鍵字搜尋是指使用檔案分享系統時，為了要搜尋特定資料項目，而輸入關鍵字的過程。由於每個人的認知不同，對於相同的資料項目可能會給予不同的命名，例如：針對某一首電影主題曲，A可能命名其為「電影名稱.mp3」，而B則命名其為「歌名.mp3」。因此，若使用傳統的點對點檔案分享系統時，系統使用者C發出對於此歌曲之查詢訊息，其關鍵字為此電影之名稱，則可能只會收到A所分享之檔案，而沒有找到B所分享之檔案。

語義關鍵字搜尋(Semantic Keyword Search)則是指在上述檔案搜尋過程中，系統會將所有的關鍵字建立起一個「關鍵字關係資料庫」，那麼當發生上述之情形，則可順利地解決找不到檔案或是有提供檔案但無法分享之情形。

語義關鍵字搜尋在主從式架構中，由於可以將語義資料庫放在中央伺服器，因此可以輕鬆的完成；而在點對點網路中，節點之間並沒有一個中央伺服器可以存取，因此要如何點對點網路中完成語義關鍵字搜尋是相當困難的。例如：Gnutella是一種相當受歡迎的非結構式點對點網路架構，由於其架構屬於分散式，因此缺乏機制記錄上述所提到的語義關鍵字關係等語義知識。而目前一般點對點檔案分享軟體，也只提供單純的「文字匹配搜尋」(Text-Match Search)。

2.3.2 關鍵字關係資料庫(Keyword Relationship Database)

2004年6月東京大學Kiyohide Nakauchi等人提出了點對點語義關鍵字搜尋的架構[11]，其概念是將系統內每一節點所分享之檔案擷取出關鍵字關係列表成為一個關鍵字關係資料庫(Keyword Relationship Database, 以下簡稱KRDB)，藉由此資料庫，將語義關鍵字的概念在點對點網路架構上實作。其系統運作流程如圖9所示：

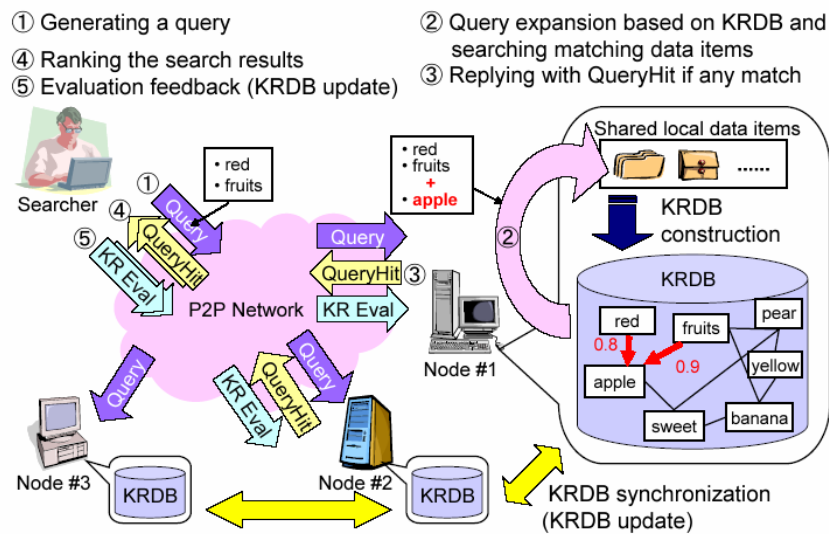


圖 9 點對點語義關鍵字查詢網路架構[11]

2-3節中曾提及目前點對點關鍵字搜尋之搜尋效率，上無法和一些先進的資訊擷取(Information Retrieval)演算法並駕齊驅，這是因為目前點對點檔案分享系統只支援單純的「文字匹配搜尋」(Text-Match Search)，而無支援語義搜尋(Semantic Search)。因此搜尋結果往往只能找到一些和之前鍵入之關鍵字有「文字匹配」的資料，這就是2-3節中所提到目前點對點網路架構所面臨的問題之一。

2.3.2.1 關鍵字關係資料庫的形成

在 KRDB，系統是完全分散式的，系統中每個節點都會根據自身所分享出來的檔案，產生出自己本地的關鍵字關係資料庫(KRDB)，其過程如圖 9。此論文假設系統中各個節點所分享出來的音樂檔案，可根據其 metadata 假設同一個檔案所提供出關鍵字有 n 個，則此 n 個關鍵字彼此都有關係。

圖 10 為系統中某一節點之 KRDB 產生過程。此節點所分享出來的檔案有三個，分別為 item1、item2、以及 item3。假設以 item1 為例子，在 item1 中擷取出四個關鍵字：A, B, C, D。系統假設這四個關鍵字彼此兩兩都有關係，則產生出六個「關鍵字關係(Keyword Relationship, 以下簡稱 KR)」，而系統給定各個 KR 一個初始值為=0.5，公式如下：

$$KR = \frac{Score^{Helpful}}{Score^{Used}}$$

$Score^{Helpful}$: 系統語義關鍵字搜尋過程中，若節點有提供相關檔案以供
 下載時，此分數將會增加一

$Score^{Used}$: 系統語義關鍵字搜尋過程中，若有真正被搜尋者下
 載時，此分數將會增加一

此架構下 KRDB 之更新分為以下兩種：

- 若 X 與 Y 且 X 與 Z 有關係，則計算其兩個 KR 的強度，若相乘之值大於系統所設定的門檻值，則增加 Y 與 Z 的 KR 到本地的 KRDB。
- 節點之間 KRDB 同步時，各節點會互相更新其 KRDB。

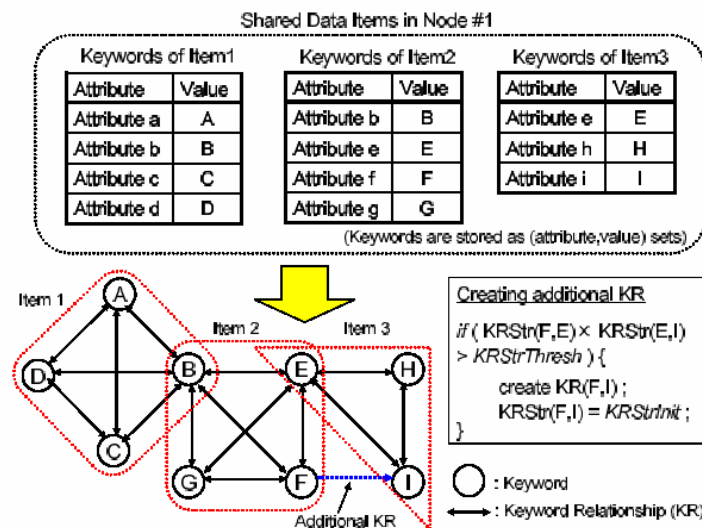


圖 10 KRDB 產生之過程[11]

2.3.2.2 關鍵字關係資料庫之評估與回饋機制

在 KRDB 架構中，KRDB 之評估回饋流程如圖 11 所示：

- 搜尋者在發出查詢要求後，一定時間之內可能會收到一個或以上的查詢命中訊息(Query Hit)，而這些發出查詢回應之節點，會在自己本身的 KRDB 上將所運用到的 KR 分數進行分數調整，此步驟如 2-4-2 小節之敘述
- 接著系統將搜尋者所收到的查詢命中訊息按照使用之關鍵字關係與原本查詢之需求中比對其關鍵字相關率及評分排名，讓搜尋者清楚地察看所有的搜

尋結果

- 最後搜尋者選擇真正欲下載之檔案，發出下載要求，而本次下載真正使用到之 KR 也需要如同 2-4-2 小節之敘述進行更新的動作

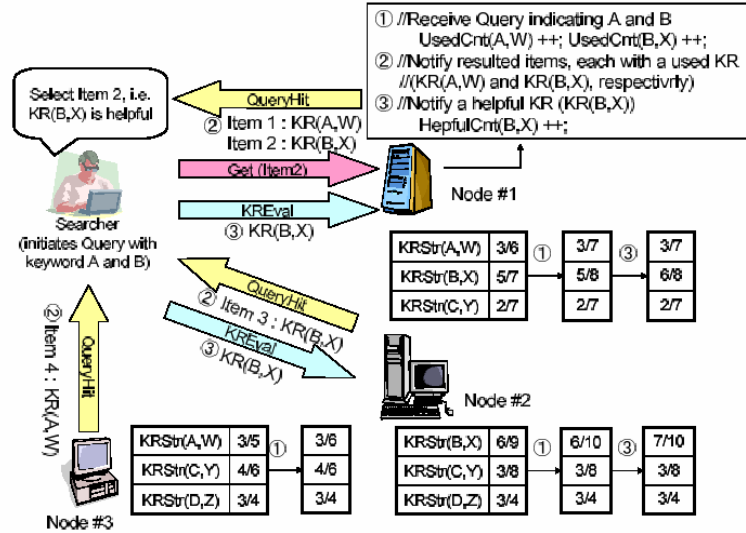


圖 11 KRDB 評估回饋之過程[11]

2.3.2.2 關鍵字關係資料庫之同步機制

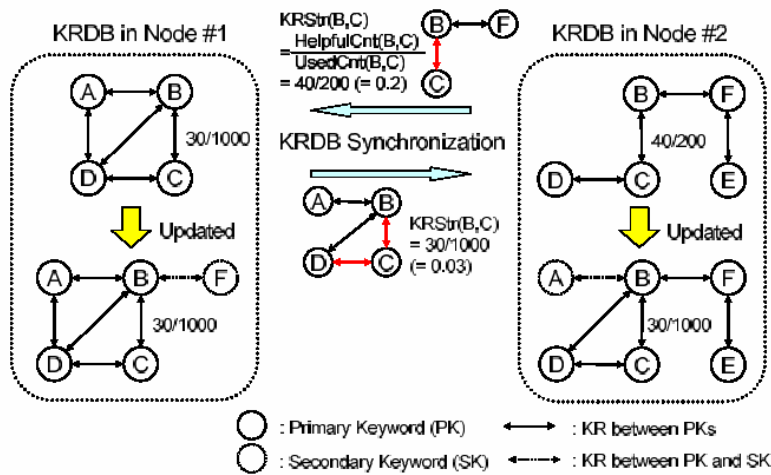


圖 12 KRDB 同步的流程圖[11]

本架構之同步方法為：系統中各個節點發出廣播訊息試圖尋找一個可同步之對象，並設定 TTL(Time to Live)以限定訊息所能傳遞之時間。若在一定時間之

內找到一個可同步之節點則進行同步。使用這種方法之缺點為：由於系統屬於完全分散式，因此分散於各分群之 KRDB 並不能夠完全同步，所達到之語義搜尋效果也就降低了。在圖 12 中，節點一(node1) 與節點二(node2)之間彼此發現，由於節點一的 KRDB 中較多 KR 是由本身所分享之檔案所產生出來的，因此節點二以節點一為同步的對象。若自己和同步對象有相同的 KR，則比較其強度分數大小，此處以 primary key 與 secondary 為其差別，若對方之 primary key 較大，則以對方之資料做同步動作；若自己沒有同步對象所擁有之 KR，則加入此 KR 到自己的 KRDB 中。



第三章 以 Super Peer 為基礎之語義關鍵字搜尋 的點對點網路

第二章中探討了一些相關的文獻，接著在第三章中將分節說明本論文提出之架構中各項機制。

3.1 需求定義

在2-4節中所提及之KRDB架構[11]，尚有許多問題仍待解決，條列如下：

- 系統架構為完全分散式，因此所有查詢要求訊息(Query Request)均無方向性地以廣播方式傳遞出去，對於系統整體的網路流量及負載平衡會有一定程度的影響。
- 在此架構下，每一節點 (Peer) 都需要保管自己的KRDB以便記錄本節點所分享之關鍵字關係，因此會造成系統資源浪費。
- 傳統的主從式架構中，有一個中央伺服器可以存放語義資料庫，但點對點檔案查詢系統中，由於架構屬完全分散式的形式，並沒有一個可以放置語義資料庫之地方，唯一的方法只有將資料庫分散於系統各處，才可以達到和原本一樣的效果。但在KRDB架構[11]裡，所有節點之KRDB同步仍無法達成真正的資料同步，因此會影響語義關鍵字搜尋之效果。

本論文將這些真正需求，提出了一個以 Super Peer 為基礎之語義關鍵字搜尋的點對點網路架構(以下簡稱為 SP-KRDB 架構)，以改善上述問題。

3.2 系統架構

本節首先介紹本論文所提出之整體架構，接著敘述架構中的各種機制。分別為「分群及 Super Peer 選定之原則」、「以 Super Peer 為基礎之 KRDB」、「KRDB 更新與同步之機制」、以及「查詢過之關鍵字的快取機制」。

本機制架構如圖 13 所示：

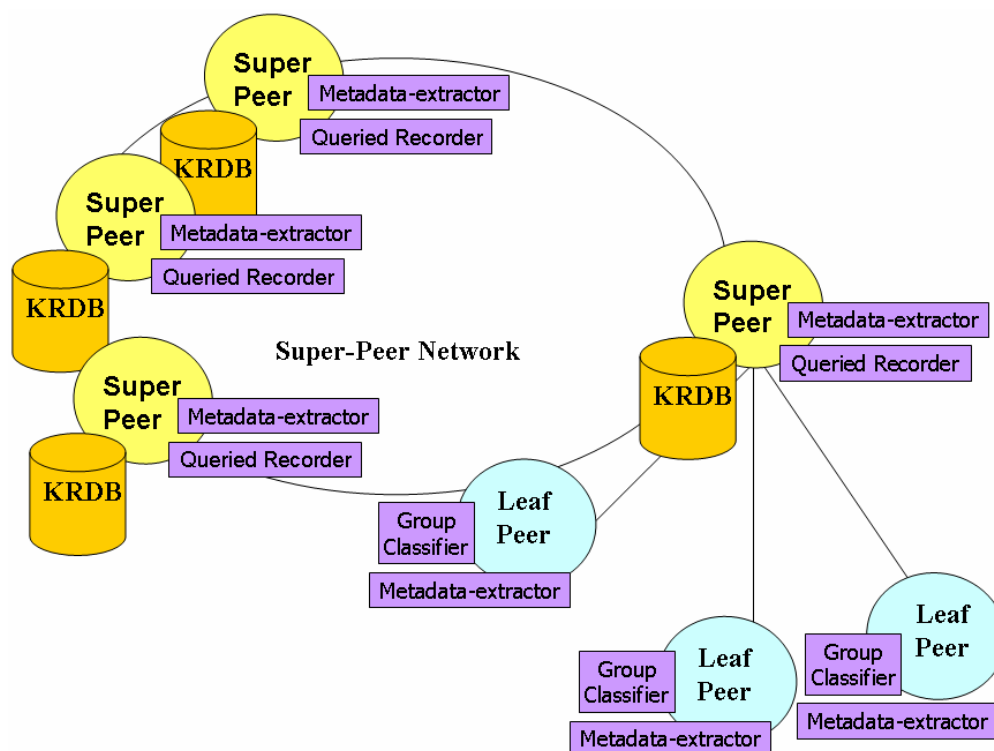


圖 13 以 Super Peer 為基礎之語義關鍵字搜尋的點對點網路架構圖

本架構所發展的機制，適用於非結構式之點對點網路架構(Unstructured P2P Network)。圖 13 為本論文所提出之架構圖，在圖中可見本架構之元件有以下幾種：

- 樹葉節點 (Leaf Peer) : 泛指系統內所有完成加入系統之節點，內含兩大元件：

- 元資料擷取器 (Metadata-extractor 以下簡稱 Metadata Extractor): 負責新加入系統之節點所分享的所有音樂資源, 擷取出其中的關鍵字關係, 然後將之回傳給 Group Classifier 所選擇之 Super Peer。
- 群組分類器(Group Classifier): 負責將新加入系統之節點, 根據其 metadata 擷取出之結果, 發出加入群組要求(Group Request), 然後選取一個合適之 Super Peer。
- 超級點(以下均簡稱 Super Peer): 泛指系統內所有被選為特定分群之領導節點(Leader Peer)。其中含有三大元件:
 - 關鍵字關係資料庫(Keyword Relationship Database, 以下簡稱 KRDB)

記錄所有非本身之樹葉節點所傳送過來之關鍵字關係 (Keyword Relationship, 以下簡稱 KR)以及整合 Super Peer 本身所分享出之 KR。
 - Queried Recorder: 將查詢過且有查詢命中(Query Hit)的訊息紀錄在各個分群之 Super Peer 上。藉由此機制, 假設當某節點 A 發出查詢要求時, 經過各個 Super Peer 的 query expansion[11], 然後以此新的查詢訊息於其 Queried Recorder Table 中尋找, 若有找到相同之查詢記錄, 則先回應 QueryHit 給搜尋者。如此一來, 可降低訊息洪流發生之機會, 且可加快熱門檔案之搜尋速度, 此點符合音樂分享檔案系統的實際需求。
- 超級點網路(Super Peer Network): 指系統中所有 Super Peer 所形成的網路, 稱為超級點網路。

本論文架構之元件如上所述。另外, 將分別將其所組成之準則以及方法分述以下小節中。

3.2.1 分群及 Super Peer 選定準則

(1) 節點分群準則

在本論文提出之架構下，新節點加入系統後，分群準則是以「距離遠近」為基準。以各個分群的 Super Peer 為中心計算 hop 數，並設定範圍在 hop 數等於某數值以內的節點，以 Super Peer 為中心，形成一個分群(Group)。至於以距離為遠近作為基準的理由是假若以此為基準，各分群內節點彼此間的距離比較近，因此在訊息傳遞的過程會比較快，且可達到網路使用量平衡。

在此假設下，根據 hop 數來分群的結果可能有以下最差之結果：

- 若 hop 數過大時，則可能造成系統中所有節點成為一個分群(Group)
- 若 hop 數過小，則可能造成系統中所有節點各自成為一個分群(Group)

因此本論文提出以下是一些模擬之假設以及前提條件：

- 本架構在非結構式的點對點網路上運作。
- 假設系統內所有節點都知道彼此的存在。
- 假設所有分群都會控制其最大加入節點數目，且所有節點均勻的分散在系統內。因此各個分群所管控之節點數目會相當平均。

上述這些假設將可使得系統達到平衡狀態，因此將以此作為本系統架構內節點分群的標準。

(2) Super Peer 選定準則

Super Peer之選定

- 先進入系統者：先進入系統之節點先當 Super Peer
- 網路速度較快者：接著選候選點(Candidate Peer)之時，根據此 Super Peer 附近條件最好者，如網路速度快之節點作為其備份之候選節點

節點尋找Super Peer之準則

- 新節點加入系統之後，發出要求加入之訊息到 Super Peer 網路去尋找 Super Peer，若在一定 hop 數之內搜尋到 Super Peer，則加入此分群
- 承上點，若搜尋到 Super Peer 數量大於 1 時，則根據其 hop 數來決定要加入哪一個分群。

譬如：搜尋到之超級點有 3 個，距離 3，4，5 hops 各有一個，則結果為加入 hops 數為 3 之分群。若搜尋到最佳加入分群之 Super Peer 有兩個以上(hop 數相同)，則隨機選擇加入分群

候選點(Candidate Peer)之選定

為避免發生某分群之 Super Peer 發生突然離開系統所造成 KRDB 遺失之現象，系統將會進行各分群 KRDB 備份之動作，備份之準則如下：

- 首先，當節點成為 Super Peer 之後，系統會設定一個門檻值 (threshold)。若分群大小在此門檻值以下則此分群之 Super Peer 需要選擇一個或是倍數之候選點 (Candidate Peer) 來為 Super Peer 做 KRDB 備份之動作，而此候選點選擇之準則為找 hops 數最近之節點

譬如：設定 threshold=10，則當分群內數量等於 10 時，Super Peer 選擇一個離自己最近之節點做 KRDB 備份；若分群內數量大於 10 時，則依其數量倍數尋找候選點做 KRDB 備份

- 在此備份機制下，若某分群之 Super Peer 發生離開系統之現象時，則原本之候選點將會升級成為 Super Peer，而此升級為 Super Peer 之節點也會尋找其他候選點為自己做 KRDB 備份

3.2.2 以 Super Peer 為基礎的關鍵字關係資料庫

在本論文提出之架構下，整合 KRDB 架構[11]與 Super Peer 網路。KRDB 架構裡各節點之 KRDB 是由各個節點字形產生、保管以及相互更新；現在則整合了 Super Peer 網路的優點，使得系統中的 KRDB 可由各分群的 Super Peer 保管，而每個分群則共用同一個 KRDB。如此一來，可增加 KRDB 同步的可能性。

本論文所提出之架構中，KRDB 設計為各分群共同擁有一個 KRDB，其保管者為各分群的 Super Peer。資料結構設計如表 1 所示：

表 1 SP-KRDB 之資料結構

rowid	Key1	Key2	Strength
1	Modonna	Take a Bow	0.5
2	Take a Bow	The Power of Love	0.7
...

其資料結構說明如下：

- rowid：此筆 KR 之編號
- Key1：此筆 KR 之關鍵字一
- Key2：此筆 KR 之關鍵字二
- Strength：此筆 KR 之強度分數



本架構中所有 KR 都是由系統中所有節點分享之檔案而來的。KRDB 架構[11]認定同一個檔案所分享出的關鍵字之間彼此均有關係，如：擷取出三個關鍵字：K1, K2, K3，則形成 (K1, K2), (K2, K3), (K1, K3), (K2, K1), (K3, K2), (K3, K1) 六個關鍵字關係(Keyword Relationship)。每個節點所分享出來之 KR 形成一個關鍵字關係列表(以下簡稱 KR List)，加入系統或是更新分享檔案時將會回傳給所屬分群之 Super Peer 保管。

本架構所設計的資料結構與 KRDB 架構[11]並無特別的差異，但由於 KRDB 是由各分群之 Super Peer 保管，因此各個分群可共用一個資料庫，減少系統資源的浪費。

3.2.2.1 KRDB 更新之機制

本論文所提出之「以 Super Peer Network 為基礎之語義關鍵字搜尋的點對

點網路架構」中，更新之機制與 KRDB 架構[11]觀念雷同，如：

- 節點更新其分享之檔案時，需要將其新增/更新之 KRList 送給所屬之 Super Peer 以利其更新 KRDB
- 節點發出搜尋且得到搜尋結果之時，會讓使用者選擇真正要下載之檔案，而此時除了發出下載要求訊息，亦會發出評估回饋訊息。此訊息係發給下載目的節點所屬之 Super Peer，告知這次所用到之 KR，以便更新其 KR 強度

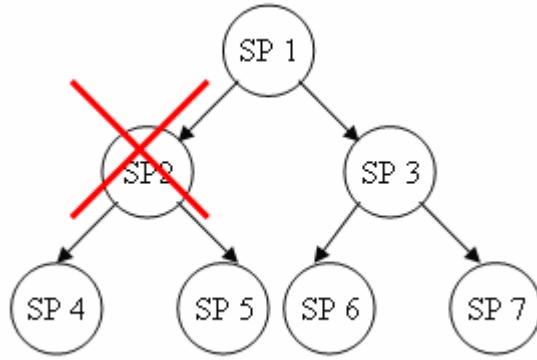
3.2.2.2 KRDB 同步之機制

在同步的方面，語義關鍵字搜尋在一般傳統的主從式架構中，由於有一集中式的伺服器可以保管整個系統中所有關鍵字關係資料庫[15]，因此系統中所有查詢訊息都可共用同一個關鍵字關係資料庫，而在非架構式點對點網路架構下，則並無一個永遠存在的點可以幫忙保管這個資料庫。

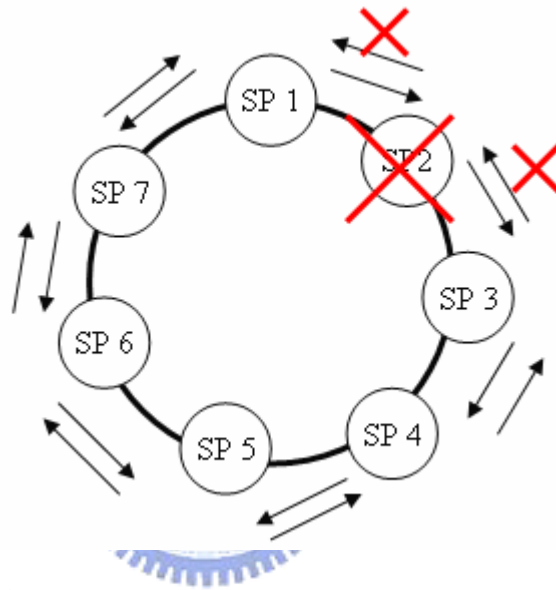
在 KRDB 架構[11]中，其 KRDB 同步的方式在第二章的文獻探討已經探討過，由於此架構屬完全分散式，其語義資料庫是由系統中每個節點保管自身所分享出來之 KRList 而形成的 KRDB；其同步的方式，則是發出廣播訊息給身邊的節點，盲目地尋找同步的對象，所以不能夠達到像以往主從式架構下的語義關鍵字搜尋的效果，因為在此篇論文中，系統中所有的點不能共享同一個語義資料庫。

在本論文架構之下，各分群之 KRDB 是由各分群之 Super Peer 所保管，因此要做到系統中 Super Peer 之間 KRDB 的同步，必須做到以下步驟：

- 本論文之 Super Peer 網路架構是以「雙環狀」的方式實現，在此是指邏輯上的雙環狀。各分群的 Super Peer 都有一個路由表 (Routing Table)，此路由表是在節點成為 Super Peer 時即擁有的，其功能為記載連接此 Super Peer 的左右鄰居節點之相關訊息，如此一來，系統中所有節點會形成一個雙環的結構。
- 在本論文中不使用「樹狀」，而採用「雙環」的原因如下所示：



(a) Super Peer Network 以樹狀方式實作



(b) Super Peer Network 以雙環狀方式實作

圖 14 Super Peer Network 架構

由於在點對點系統中強調的是系統中任何一點發生忽然消失的情況之下仍可正常運作，因此在本論文的架構下，必須讓系統在任何節點消失時也可以正常運作。圖 14 (a)是以樹狀的方式串連，若 Super Peer(以下簡稱 SP)2 消失不見，則 SP4, SP5 與系統其他點會切斷連線；而圖 14(b)則是以雙環狀的方式串連，節點之間互相連結，形成順/逆時鐘方向的雙環串連，若 SP2 消失，最先察覺其消失的節點必為其鄰居 (SP1 與 SP3)。假設 SP1 首先發現 SP2 消失，則會順著逆時針方向找尋另外一端的節點，最後找到 SP3 的右節點不見，因此 SP1 會與 SP3 建立鄰居之關係。

依照圖 14 之架構實作之，仍有可能發生 Super Peer Network 切斷的可能。在上例中：SP1 首先發現 SP2 消失，SP1 欲找尋 SP3 之時，若

發生沿途節點消失之現象，則當 SP1 偵測到環狀切斷之時(如：SP1 發現 SP4 之右節點消失但其右節點不是 SP3)SP4 會發出廣播以尋找附近之鄰居節點試圖回復 Super Peer Network 之完整，而由於系統中所有 Super Peer 在加入 Super Peer Network 時，是依據距離之遠近來加入系統，因此附近之鄰居節點必為距離較近之節點，也因此，這種類型之廣播訊息會很快地找到所要尋找的節點。

- Super Peer 間定期同步 KRDB 之資料

節點之資料結構如表 2：

表 2 節點的資料結構

Peer ID	IP Address	Role	SP_IP	LeftSP	RightSP
254	192.168.1.13	GP	192.168.1.46	Null	Null

Peer ID	IP Address	Role	SP_IP	LeftSP	RightSP
254	192.168.1.13	SP	Null	192.168.2.220	192.168.3.289

此基本資訊表之欄位說明如下：

- ◆ PeerID(節點編號)：此節點之編號
- ◆ IP Address(IP 位址)：此節點之 IP 位址
- ◆ Role：此 Peer 之角色，包括 SP(Super Peer)、CP(Candidate Peer)、與 GP(General Peer)
- ◆ SP_IP：所屬 Super Peer 之 IP 位址
- ◆ LeftSP(左鄰居節點)：若為 Super Peer，則記載左鄰居節點之資訊
- ◆ RightSP(右鄰居節點)：若為 Super Peer，則記載右鄰居節點之資訊

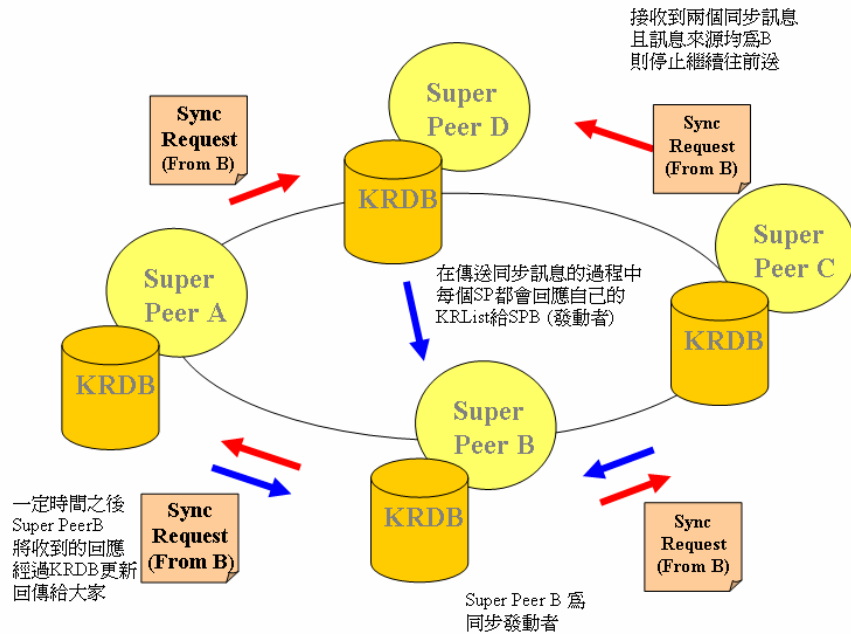


圖 15 KRDB 同步流程

- KRDB 同步之狀況有以下兩種：

(1) KRDB 初始同步：

系統在剛開始只有第一個 Super Peer，則第二個加入之 Super Peer 需要作為同步發動者，而此發動者將會執行

(2) KRDB 一般同步：

系統開始運作之後，當新的 Super Peer 加入 Super Peer Network 時，此新的 Super Peer 會發動同步(意即此新加入之 Super Peer 成為此次 KRDB 同步的發動者)；另外若某 Super Peer 執行了 KRDB 更新的動作，由於系統中最先發現其 KRDB 有更新之節點必為節點自己，因此由此 Super Peer 發動同步

- 同步之步驟如下所示：

步驟一：

同步發動者依照兩個不同方向發出同步需求訊息(Synchronize Request)，在同步訊息傳送過程中，每個 Super Peer 都會回應自己的 KRDB 給發動者以圖 15 為例，發動者為 Super Peer B。

步驟二：

當 Super Peer Network 中任何一節點接收到兩個同步訊息，且

來源均為發動者，則同步訊息停止往前送。

步驟三：在一定時間之後，發動者收到所有節點的回覆後，更新其 KRDB 且回傳最新之 KRDB 給所有節點。

3.2.2.3 搜尋結果評分排行之機制

本論文所提出之架構中，由於搜尋者所得到之搜尋結果可能會超過一定數目，因此需要一個搜尋結果評分及排行之機制供搜尋者做為參考。至於本系統架構評分之機制，是根據搜尋結果所使用到之關鍵字有多少來決定的。

本系統架構中設定其搜尋結果分數範圍如下：

$$0 < \text{搜尋結果分數} < 5$$

其評分之公式為：

$$\text{搜尋結果分數} = \frac{\text{搜尋結果使用之關鍵字數目}}{\text{搜尋擴展之關鍵字數目}} \times 5$$

3.2.2.4 查詢過的關鍵字之快取機制

基於 3.2.2 小節所提及之機制，將 Super Peer 網路整合到原有的 KRDB 架構上，但在各分群內，為了讓分群內不至於因為訊息洪流而影響搜尋效果，因此本論文提出了一個「查詢過的關鍵字之快取機制」。此機制將查詢過且有查詢命中的相關訊息紀錄在各分群之 Super Peer 上。紀錄之資訊包括：此查詢命中之節點編號(PeerID)以及相關資訊。如此一來，每當各個分群之 Super Peer 在接收到任何查詢訊息時，可以先搜尋本端之查詢紀錄資料庫，這樣可以減少各分群內之訊息流量，以達成降低訊息洪流發生之機會。其資料結構如下：

表 3 查詢記錄表之資料結構

PeerID	Keywords	Network_Type
254	Taitanic,鐵達尼	ADSL
729	My Heart Will Go On ,Taitanic	T1
914	Somewhere Out There ,美國鼠譚	T1
...
...

- PeerID：此筆查詢紀錄之節點編號

- Keywords：相關之關鍵字
- Network_Type：節點之網路類型

步驟一：

搜尋者發出查詢訊息，若有節點擁有相關檔案，此節點所屬之 Super Peer 會將此查詢命中之相關訊息紀錄在其上「查詢記錄表」(Queried Recorder Table)上。

步驟二：

當之後有其他節點也要搜尋相似之檔案時，若經過 Super Peer 中 KRDB 之查詢擴展後之「新查詢訊息」，在其「查詢記錄表」中能夠找到相同之記錄，那麼就可以先行回應搜尋點，告知哪一節點擁有相關檔案。



3.3 系統運作流程

本小節將介紹本論文所提之架構的系統運作流程(見圖 16)，包括：新節點加入系統、已加入系統節點更新其分享之檔案、查詢過程、搜尋結果排名、搜尋結果回饋、KRDB 同步、以及節點離開系統等。

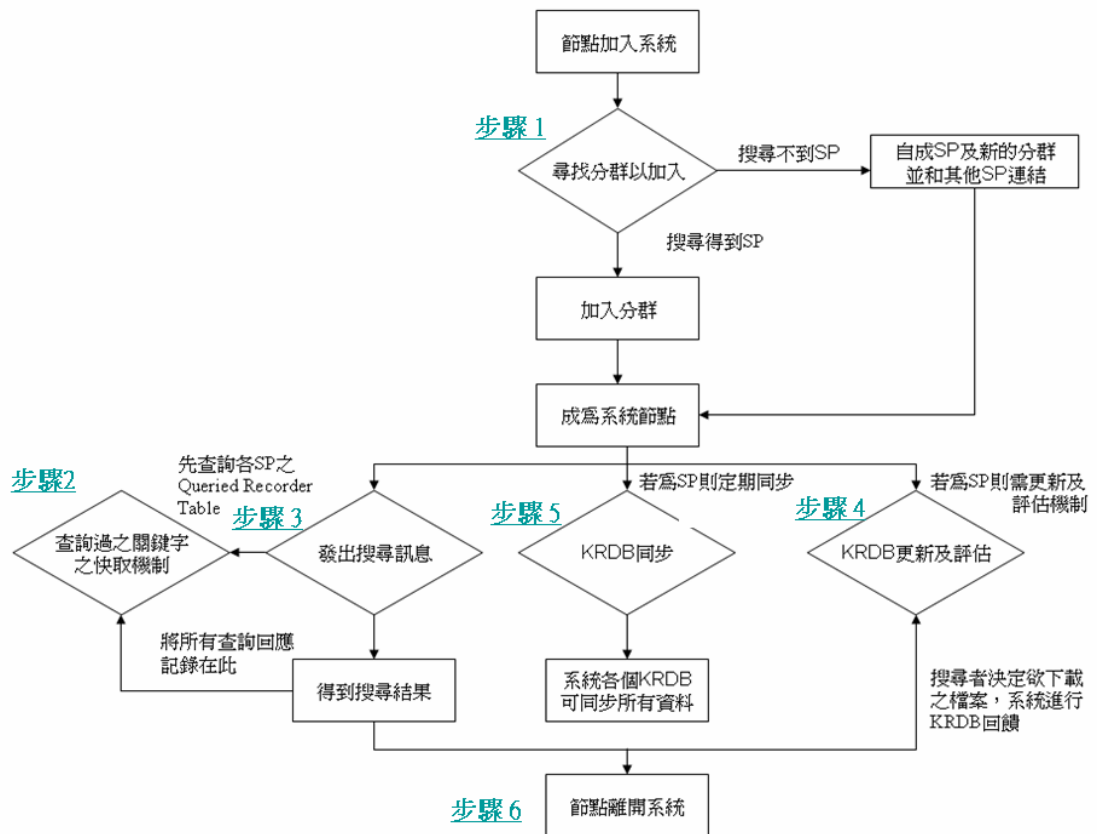
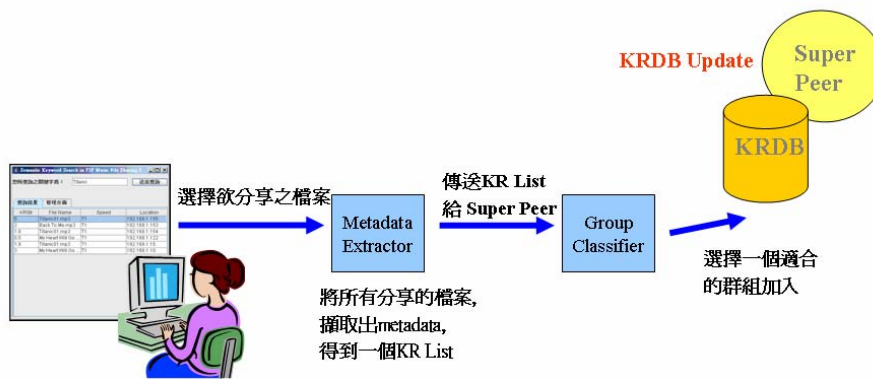


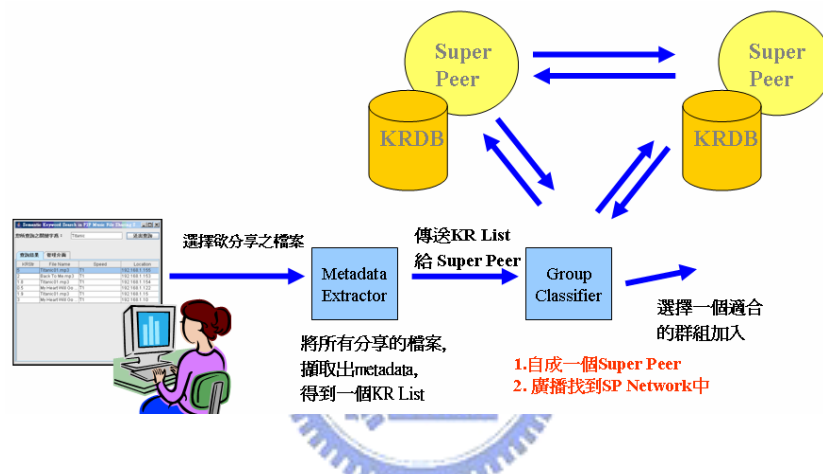
圖 16 系統運作流程圖

3.3.1 新節點加入系統

在本架構下，新節點若要加入系統，其程序圖如下：



(a)在一定範圍內尋找到分群加入



(b)在一定範圍內沒有尋找到分群加入

圖 17 新的節點加入系統

在圖17中，分為以下兩種情況：

情況一：在一定範圍內尋找到分群加入

1. 新的節點加入系統，首先成為系統其中一個樹葉節點；使用者選擇欲分享之檔案；接著Metadata Extrator將會根據其分享之檔案，擷取出關鍵字關係列表(KR List)。
2. 群組分類器將會根據論文中第三章所提及之節點分群準則，為節點找到一個適合的群組加入。
3. 最後將剛才所提到之KR List送給此分群之Super Peer。

情況二：在一定範圍內沒有找到分群加入

1. 新的節點加入系統，首先成為系統其中一個樹葉節點；使用者選擇欲分享之檔案；接著Metadata Extrator將會根據其分享之檔案，擷取出關

鍵字關係列表(KR List)。

2. 群組分類器將會根據3.2.1小節提及之節點分群準則，為節點找到一個適合的群組加入。
3. 若在一定範圍內沒有找到Super Peer則自成一個Super Peer，然後廣播找到Super Peer Network。

3.3.2 已加入系統之節點更新其分享檔案

在本架構下，已加入之節點欲更新其分享之檔案，其程序圖如下：

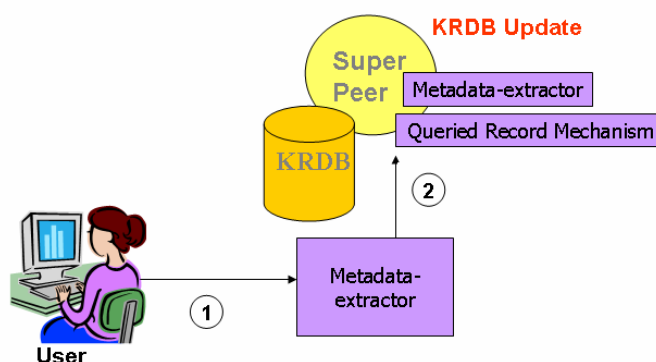


圖 18 已加入之節點欲更新其分享之檔案

在圖 18 裡，分為以下步驟：

1. 已加入之節點欲更新其分享之檔案，Metadata Extrator將會根據其分享之檔案，擷取出關鍵字關係列表(KR List)。
2. 將剛才所提到之KR List送給此分群之Super Peer。

3.3.3 節點送出查詢訊息、查詢命中、搜尋結果排名

當系統某一節點發出查詢訊息，所因應而生之事件，其程序圖如下：

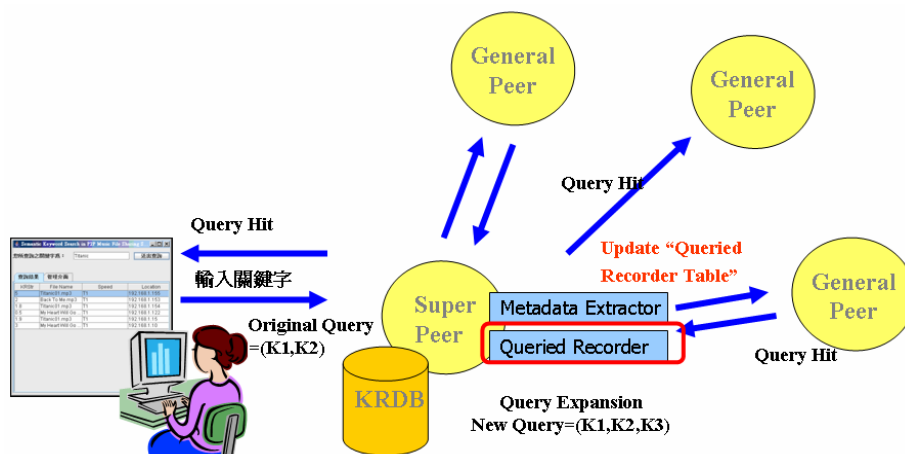


圖 19 節點送出查詢

在圖 19 裡，分為以下步驟：

1. 節點送出查詢訊息，假設其夾帶欲查詢之關鍵字為K1, K2，此訊息首先會被送到本身所屬之Super Peer，此作法之理由為：由於先送到距離自己較近之分群內做搜尋，因此所回覆之速度會更快，而當一定時間之後，此查詢會由此所屬之Super Peer再分散出分群外給各分群之Super Peer，接著再做以下的搜尋動作。
2. 假設某一Super Peer接收到此查詢訊息，則根據內部之KRDB做查詢擴展之動作，舉例如下：
 - 在此KRDB中，根據[11]中的Query Expansion方法，原本的查詢變成(K1, K2, K3)
 - 接著用新的查詢訊息(K1, K2, K3)，到Queried Recorder Table中尋找，若有搜尋到則代表以前曾經有查詢過，則告知搜尋者哪一個節點上有這些keyword的相關資料
 - 接著Super Peer會將查詢訊息往分群內送出，一定時間內若有節點回覆則傳遞一個查詢命中(Query Hit)訊息給搜尋者，且將此經過Query Expansion之查詢訊息及發現有相關資料之節點記錄在Queried Recorder之Table中

3.3.4 評估回饋及記錄

當系統中某一節點發出查詢訊息，所因應而生之事件，其程序圖如下：

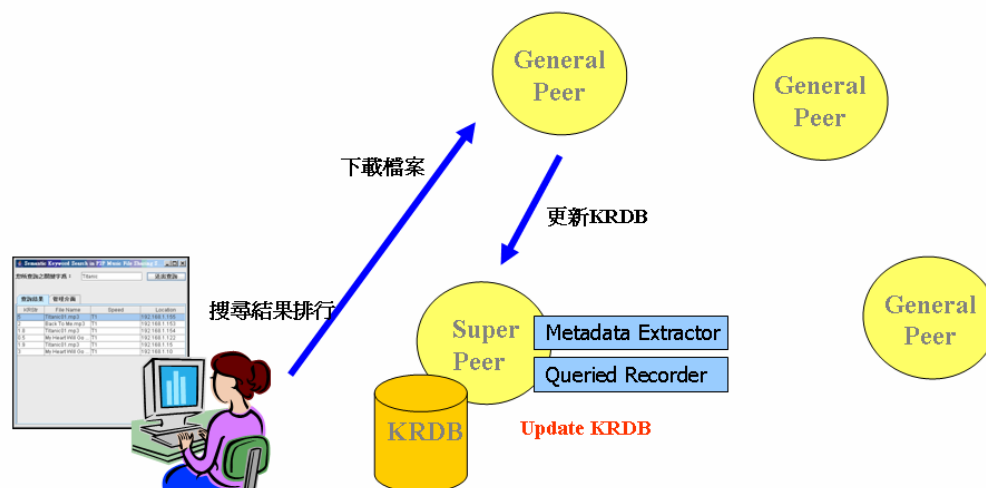


圖 20 評估回饋及記錄

在圖 20 中，分為以下步驟：

- 搜尋者在一定時間之後，可能會接收到許多查詢命中(Query Hit)回應，系統會根據本論文中搜尋結果評比之機制(見3-4-3)，給定各個回覆訊息分數，再交由搜尋者自行選擇下載。
- 當搜尋者點選下載檔案之後，系統會自動評估回饋到KRDB[11]，其作法係將此次使用到之KR在KRDB中之強度增加
- 接著搜尋者開始直接下載此檔案

3.3.5 KRDB 同步

KRDB 同步，其程序圖如下：

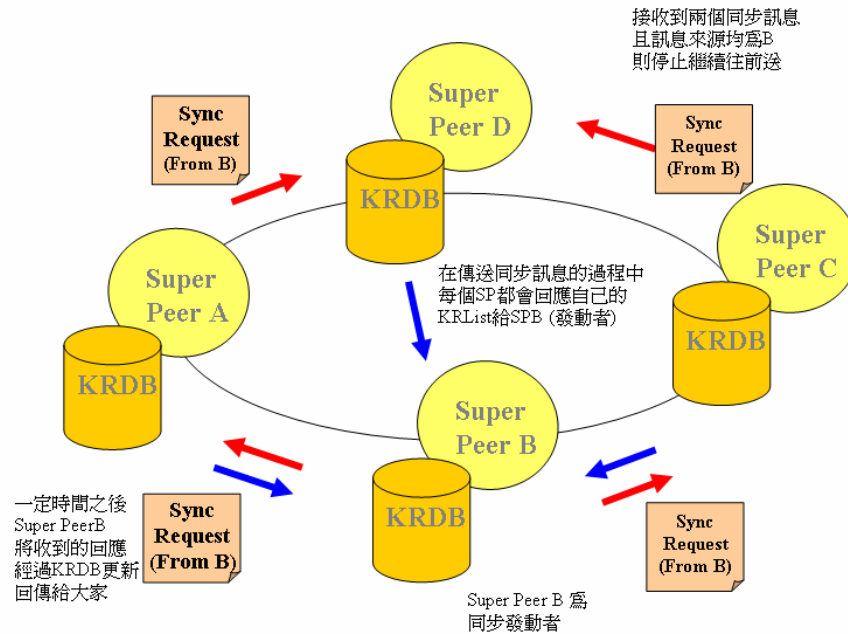


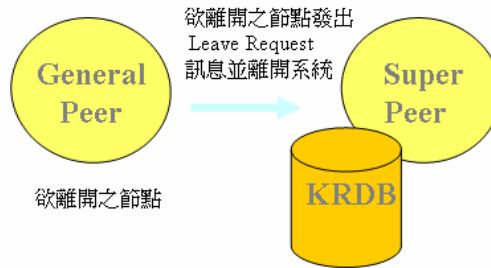
圖 21 KRDB 同步

圖 21 中，分為以下步驟：

- 同步發動者之選取，如 3.2.2.2 小節所述
- 同步發動者依照兩個不同方向發出同步需求，訊息傳送過程中，每個 Super Peer 都會回應自己的 KRDB 給發動者
- 當 Super Peer Network 中任何一節點接收到兩個同步訊息，且來源位址均為發動者，則同步訊息停止往前送
- 在一定時間之後，發動者收到所有節點的回覆，更新其 KRDB 且回傳最新之 KRDB 給所有節點

3.3.6 節點離開系統

節點離開系統分為以下兩種情況，其一之程序圖如下：



PeerID	K1	K2	Network_Type
uuid-59616261646162614A7874615032503303958F2E507B494CBBDA9576E1F6FA7E03	Taitanic	鐵達尼	ADSL
uuid-6261646162614303958F2E507B494CBBDA9576E1F6FA7E03A75961874615032503	My Heart Will Go On	Taitanic	T1
uuid-F2E5076261646162614303958B494CBBDA9576E1F6FA75032503E03A7596187461	Somewhere Out There	美國鼠譚	T1
...
...

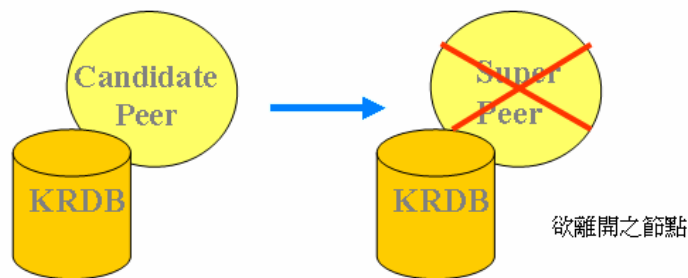
查詢紀錄表 (Query Recorder Table)

圖 22 分群內一般節點離開系統

在圖 22 中，分為以下步驟：

- 系統中一般節點若要離開系統，所屬之 Super Peer 偵測到其離開，則會在其查詢紀錄表格中刪除此節點之相關資料
- 此節點即刻離開系統

第二種情況為分群內 Super Peer 離開系統，其程序圖如下：



Super Peer 突然從系統離開，此時分群中之候選點偵測到 Super Peer 的離開之後，即刻啟動成爲此分群之 Super Peer

圖 23 分群內 Super Peer 離開系統

圖 23 中，分為以下步驟：

- 若分群之 Super Peer 要離開系統，此時分群中之候選點會偵測到其離開，即刻成爲此分群之 Super Peer
- 另外系統中其他的 Super Peer 要更新其 Routing Table

接著為第三種情況為為分群內 Super Peer 與候選點均離開系統，其程序圖如下：

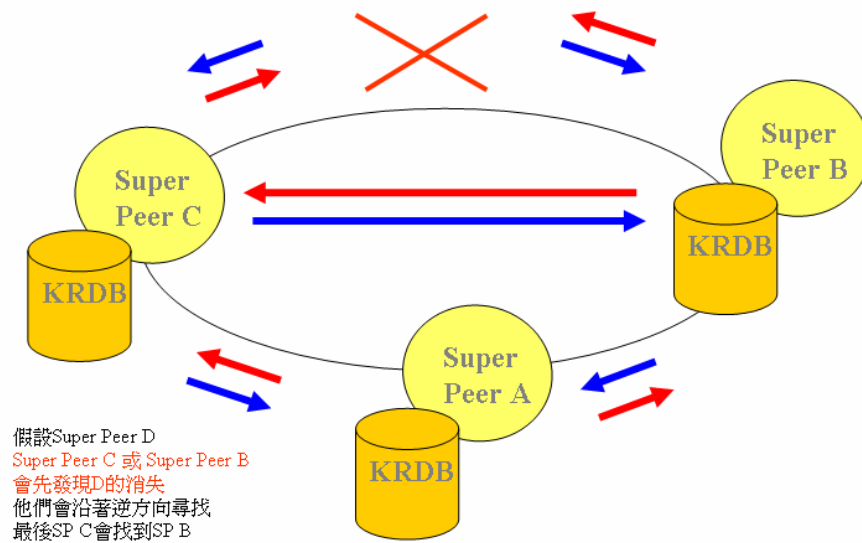


圖 24 分群內 Super Peer 與候選點均離開系統

在圖 24 中，分為以下步驟：

- 首先，若分群內發生 Super Peer 與其候選點均消失離開系統之情況，系統內最早發現其消失之節點必為此節點之鄰居，在本圖例中，Super Peer(簡稱 SP)D 消失最先發現的節點為 SPC 與 SPB。由於本系統架構為「雙環狀」之架構，因此每個節點都會記載著左右鄰居節點之資訊，也因此，當某一個連結被切斷時，可以沿著逆方向去尋找節點以要求回復串連。在本架構中，假設 SPC 先發現 SPD 的消失，則 SPC 沿著逆方向發出連結要求之訊息
- 當 SPB 收到此訊息，且發現 SPC 之右鄰居節點為 SPD，SPB 將會與 SPC 互相連結起來，如此一來，可避免 Super Peer Network 遭切斷

第四章 模擬實作與成果分析

-以音樂檔案分享為例

在第三章中已將本論文所提出之架構詳細說明過，在本章節裡，將根據進行模擬實作，模擬的目的在於驗證採用本論文所提出之架構，在搜尋效能(包括：搜尋準確度、搜尋速度等等)上是否有比 KRDB 架構[11]好。本章將依序說明本系統之系統環境、模擬方法、實驗假設、以及成果分析。

4.1 系統環境

本論文架構之模擬實驗環境，如下所示：

硬體：個人電腦一台

CPU：Intel Pentium M 1.5G，RAM：512M，硬碟空間：160G

軟體：作業系統為 Windows XP Professional，模擬實作語言：JAVA 2



4.2 模擬實作

4.2.1 系統設計

本論文模擬實作之系統是採用 Java 語言實作。整個系統主要分成五大模組，茲分述如下：

- 介面模組
除了實作系統介面外，還包括了一些系統初始設定，與輸出設定等
- Super Peer 模組
實作 Super Peer 的模組。其功能包括分群機制、Super Peer 選定、KRDB 機制的功能
- General Peer 模組
實作 General Peer 的模組。其功能包括分群機制、尋找最近 Super Peer、更新 KRDB 機制
- Super-Peer-based KRDB 環境模組
主要實作本文所提的基於 Super-Peer-based KRDB 的機制
- Queried Recorder 環境模組
主要實作出本文所提的查詢過之關鍵字快取機制

4.2.2 模擬假設

本系統架構有數項假設，茲條列如下：

1. 各個節點所分享的檔案均按照 ID3 tag 格式存在[21]，因此所有分享出來的關鍵字關係都是有意義的。
2. 所有節點互相知道彼此的存在。
3. 各個節點所分享的檔案中，各個檔案之 metadata 均互相有關係，因此可以形成一個有意義的語義資料庫-「關鍵字關係資料庫」。
4. 所有查詢過程以及查詢回應過程中沒有發生錯誤情況(指網路層會幫忙處理)。
5. 某分群內之 Super Peer 消失並離開系統，其分群內之候選點可立即偵測及成為 Super Peer，並通知左右鄰居節點，可避免 Super Peer Network 遭切斷。
6. 每個回應與查詢都是即時且立即收到，也就是假設沒有媒介延遲與處理延遲。

4.2.3 模擬實作之架構

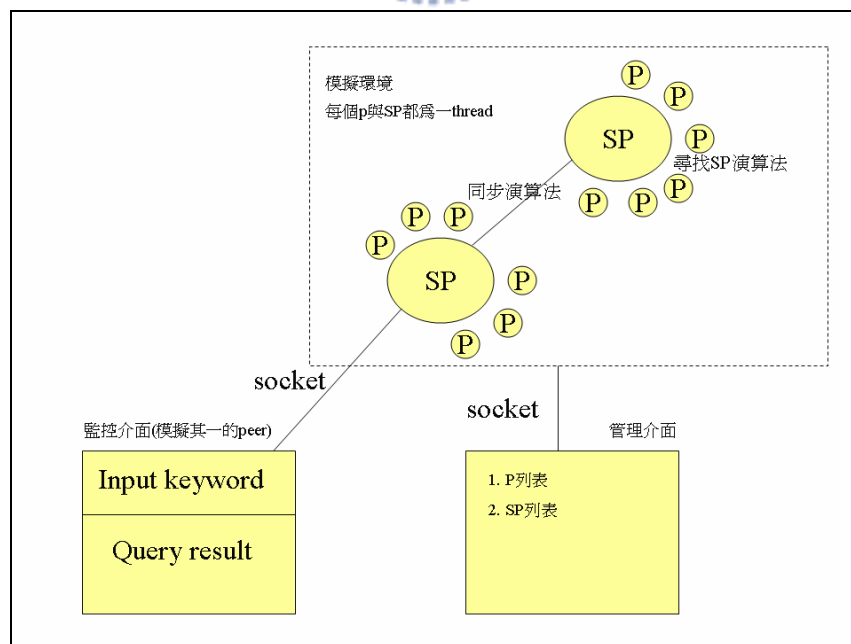


圖 25 為本模擬實作之系統架構圖，本系統以 JAVA 之 thread 模擬點對點系統中之各個點(Peer)。圖右上方為模擬之系統架構，所有節點分群管理，節點之間有搜尋 Super Peer 之演算法以及 Super Peer 之間同步演算法。

4.2.4 模擬實作之類別說明

本論文之模擬實作如 5.1 節所提到為使用 JAVA 程式語言撰寫之。本小節將系統模擬之類別細述如下：

- 程式部分：主程式操控所有類別之運作，包括模擬節點等等，在此，將以樹狀的方式呈現。如圖 26 所示：

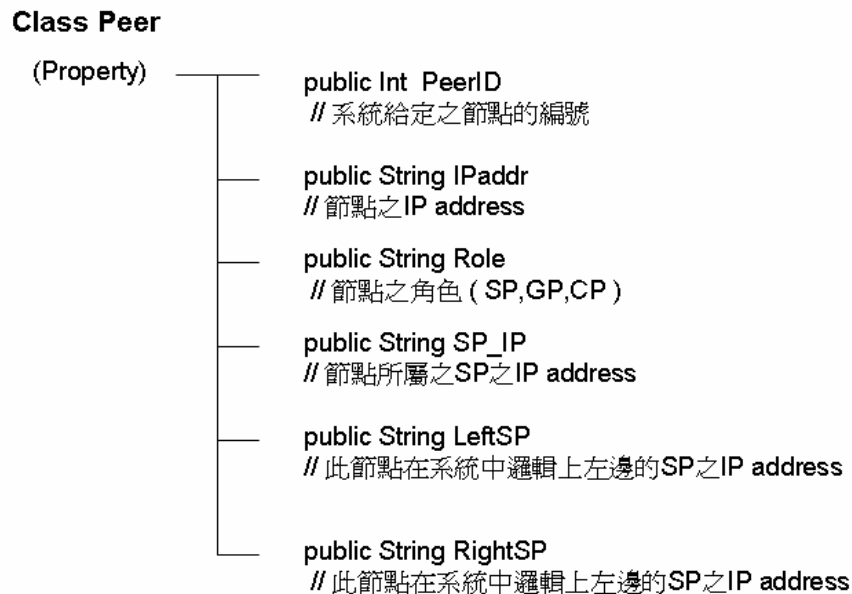
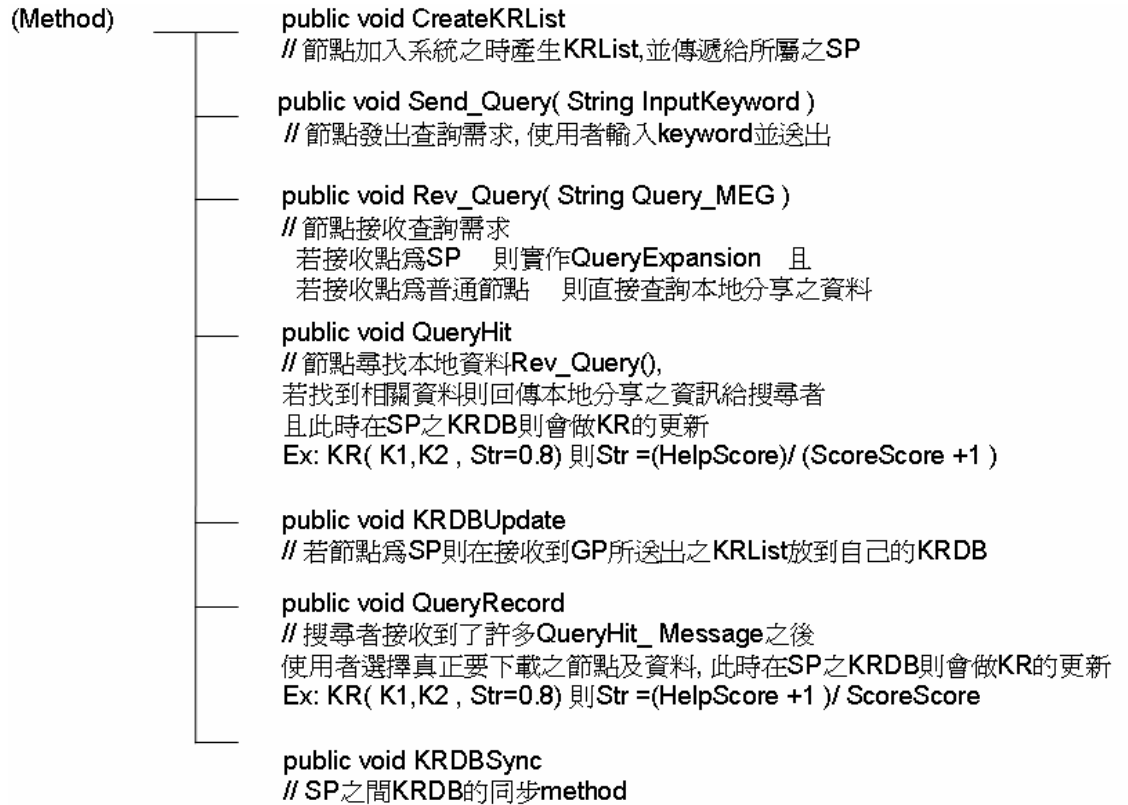


圖 26 模擬程式功能樹



註：Peer的method有些是SP才需要用到,有些是所有Peer都會用到, 如下所示

```

SP method – // ( Peer.isSP = True )
KRDBUpdate();
KRDBSync();
QueryRecord();

All Peers method –
CreateKRList ();
Send_Query();
Rev_Query();
QueryHit();

```

圖 26 模擬程式功能樹(續)

圖26為本模擬實作之節點類別(Peer Class)，圖中將所有會使用到之類別屬性以及一些會使用到之方法等等記錄在節點說明中(註解為各項之說明)。

4.3 系統畫面

本系統之畫面如下所示：



圖 27 查詢畫面

如圖 27 所示，本系統提供「監控」與「管理」介面：在「查詢結果」中，使用者可以根據自己所需要的檔案，輸入關鍵字以利查詢。圖 27 之例子為：使用者想要搜尋電影「鐵達尼號」之主題曲，但由於忘了此首歌曲之名稱，所以鍵入電影名稱“Titanic”以搜尋之。透過語義關鍵字關係資料庫(KRDB)，可以找到系統中相關的檔案，這些擁有相關關鍵字的檔案會一一列表在查詢結果中。在本例子中，使用者可以查詢得到許多搜尋結果，而這些搜尋結果會根據其所使用到之 KR 強度給定分數及排行，使用者將可看到每一筆資料的關鍵字關係強度、檔案名稱、節點網路速度、以及其 IP Address。

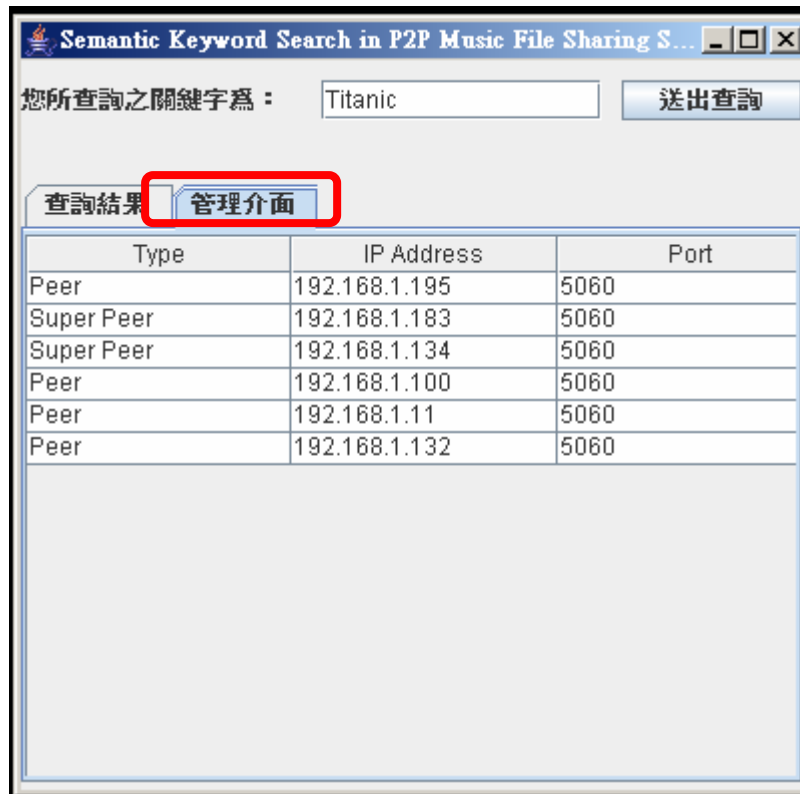


圖 28 管理畫面

如圖 28「管理畫面」中，可看到節點之列表，而這些節點之列表之相關資訊包括「節點類別」、「節點 IP Address」、以及「節點所使用之 port」。按下切換頁面即可回到「查詢結果」頁面。

4.4 成果分析

本節將會分析本研究架構所提出之方法是否可以在點對點環境中達到較高的語義關鍵字搜尋效果。首先，先簡單說明成果評估的方式。本論文將根據以下兩個指標進行成果分析，分別為：metadata 正確率與成功率。

(a) Metadata 正確率(Metadata Correctness)

Metadata 正確率(P_i):本論文以 Metadata 正確率作為評估準則。 P_i 代表此方法架構下所能搜尋到的目標檔案數量與傳統點對點檔案搜尋系統所能搜尋到的目標檔案數量之比例。在此，目標檔案指擁有使用者欲尋找之內容 i 的檔案。 P_i 越高則代表此方法架構所提供之 metadata 資訊越充足，這也代表了，此方法架構將可達到較高的語義關鍵字搜尋效果，且其效果將會趨近於集中式語義資料庫所達到的效果。以下為本研究架構(以下簡稱 SP-KRDB)以及 KRDB 架構(以下簡稱 KRDB 架構)的 P_i 表示式。(公式(4-1) (4-2) 之 Metadata 正確率公式推導與計算過程，詳見附錄 A)。

(1) KRDB 架構：

$$P_i(KRDB) = 1 - \frac{N_i^{Org}}{N_i^{KRDB}} \quad (4-1)$$

N_i^{Org} : 傳統點對點檔案搜尋系統所能搜尋到之目標檔案數量

N_i^{KRDB} : 以 KRDB 方法架構所能搜尋到之目標檔案數量

(2) SP-KRDB 架構：

$$P_i(SP - KRDB) = 1 - \frac{N_i^{Org}}{N_i^{SP-KRDB}} \quad (4-2)$$

N_i^{Org} : 傳統點對點檔案搜尋系統所能搜尋到之目標檔案數量

$N_i^{SP-KRDB}$: SP-KRDB 方法架構所能搜尋到之目標檔案數量

$$\text{又 } P_i = 1 - \frac{N_i(X_i \cap Y_i \cap Z_i)}{N_i(X_i \cup Y_i \cup Z_i)} \quad (\text{詳見附錄 A})$$

$N_i(X_i \cap Y_i \cap Z_i)$: 在傳統點對點檔案搜尋系統中，若要尋找和 x, y, z

均相關的檔案時，所能搜尋到之目標檔案數量

$N_i(X_i \cup Y_i \cup Z_i)$ ：系統中所有包含在 X、Y、以及 Z 集合中所有語義關鍵字搜尋系統可搜尋到的目標檔案數量

(b)成功率 (Success Ratio)：

成功率($P_{success}$)指在系統中所有存在之目標檔案數量與此方法能夠找到的目標檔案數量之比例。

(1)KRDB 架構：

$$P_{success}(KRDB) = \frac{N_i^{can\ be\ found\ using\ KRDB}}{N_i^{existed\ in\ the\ system}} \quad (4-3)$$

$N_i^{can\ be\ found\ using\ KRDB}$ ：KRDB 方法架構所能搜尋到之目標檔案數量

$N_i^{existed\ in\ the\ system}$ ：傳統點對點檔案系統所能搜尋到之目標檔案數量



(2)SP-KRDB 架構

$$P_{success}(SP - KRDB) = \frac{N_i^{can\ be\ found\ using\ SP - KRDB}}{N_i^{existed\ in\ the\ system}} \quad (4-4)$$

$N_i^{can\ be\ found\ using\ SP - KRDB}$ ：SP-KRDB 方法架構下所能搜尋到之目標檔案數量

$N_i^{existed\ in\ the\ system}$ ：傳統點對點檔案系統所能搜尋到之目標檔案數量

根據以上公式可得知，當 P_i 越高，代表所能找到的目標檔案越多；而根據公式(4-3)(4-4)可推論，由於所能尋找到目標檔案變多了，因此 $P_{success}$ 也會隨之提高，因此本節之實驗目的，即證明本論文所提出之架構的 $P_{success}$ 比 KRDB 架構來得高。

本節中，將由 4.3 節模擬實驗所得出之數據資料帶入公式進行分析。如前述，實驗中將欲找到之檔案稱為「目標檔案」，而為了要找到這些目標檔案，實

驗裡必須輸入一些關鍵字進行搜尋。本次實驗分別模擬了三次搜尋檔案的過程，其使用之關鍵字如下所示：

- ◆ Keyword x = "Celine Dion"，選定歌手名稱當作第一次搜尋檔案所用之關鍵字
- ◆ Keyword y = "My Heart Will Go On"，選定歌曲名稱當作第二次搜尋檔案所用之關鍵字
- ◆ Keyword z = "Titanic"，選定此目標檔案其他名稱（在本實驗中使用其電影名稱）當作第三次搜尋檔案所用之關鍵字

表 4 為本模擬實作之實驗數據，分別為 KRDB 架構以及本論文提出 SP-KRDB 架構。三次之實驗結果如表 4 所示：

表 4 搜尋結果實驗之數據

	Exp.#1 (KRDB)	Exp. #1 (SP-KRDB)	Exp.#2 (KRDB)	Exp. #2 (SP-KRDB)	Exp.#3 (KRDB)	Exp. #3 (SP-KRDB)
$Ni(Xi \cap Yi \cap Zi)$	12	12	11	11	12	12
$Ni(Xi \cap Yi)$	45	70	38	72	56	68
$Ni(Yi \cap Zi)$	8	32	6	22	9	35
$Ni(Xi \cap Zi)$	1	10	2	15	2	15
$Ni(Xi \cap \bar{Yi} \cap \bar{Zi})$	0	11	0	4	0	2
$Ni(\bar{Xi} \cap Yi \cap \bar{Zi})$	5	12	8	23	3	25
$Ni(\bar{Xi} \cap \bar{Yi} \cap Zi)$	2	16	6	20	7	18
Pi	0.84	0.93	0.84	0.95	0.86	0.94
$Pi(x)$	0.20	0.30	0.28	0.31	0.21	0.32
$Pi(y)$	0.54	0.67	0.52	0.69	0.50	0.68
$Pi(z)$	0.68	0.76	0.65	0.78	0.66	0.75

根據以上數據資料，可計算出兩種架構之 metadata 正確率，其比較圖如圖 29：

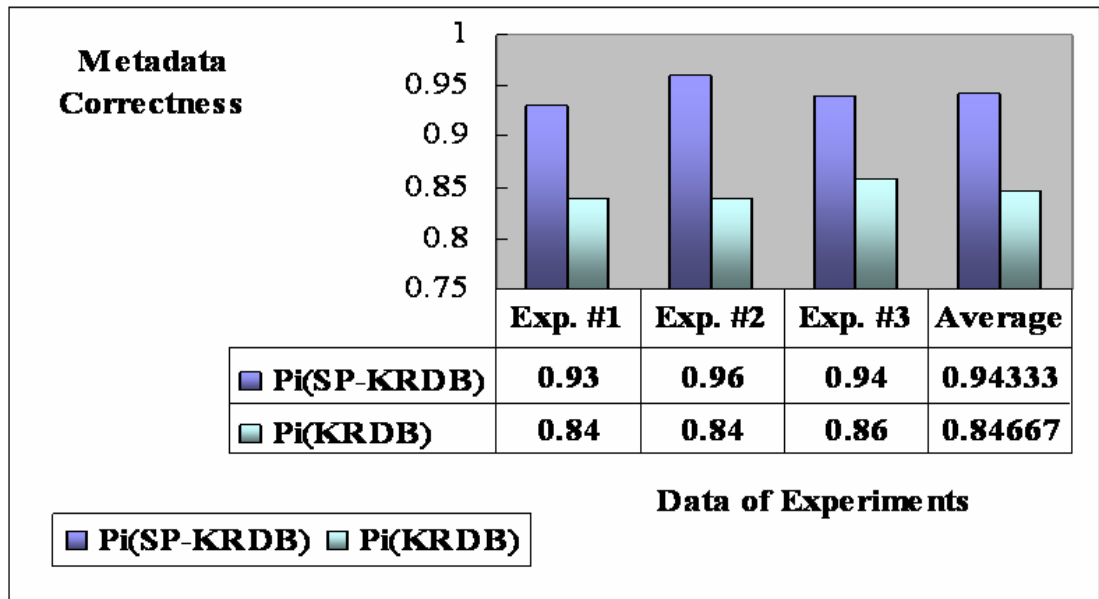


圖 29 實驗之 metadata 正確率

由圖 29 可得知本論文所提出之 SP-KRDB 架構之 P_i 平均而言比起原 KRDB 架構提高了約 11%，代表本論文之 SP-KRDB 架構之 $P_{success}$ 高於 KRDB 架構。因此可知本論文所提出之架構的確可以提升在點對點網路環境中語義關鍵字搜尋之效果，這也代表了本系統架構可以的確可以改善原 KRDB 架構中 KRDB 同步之問題，進而使得語義關鍵字搜尋在點對點網路環境中可以達到和主從式架構中，集中式資料庫相同之效果。

4.5 與以往方法之比較

本節將會進行與以往方法比較的部份。本系統架構(SP-KRDB 架構)改進了許多 KRDB 架構之缺點，本文將以表格方式以及補述方式對於本架構和以往之方法做比較。

表 5 為 SP-KRDB 架構與 KRDB 架構之比較表：

表 5 SP-KRDB 架構與 KRDB 架構比較表

比較項目 \ 架構	KRDB 架構	以 Super Peer 為基礎之 KRDB 架構 (簡稱 SP-KRDB 架構)
訊息傳遞方式	盲目發送廣播訊息	有方向性地傳遞訊息
KRDB 同步	困難	簡易
KRDB 儲存空間	系統中每一個節點 都需要保管本身的 KRDB	各分群之 Super Peer 需要保管 各分群之 KRDB

- 訊息傳遞方式：

原本之架構由於整個系統所有之訊息都是用盲目廣播的方式，譬如：某節點發出查詢訊息，則設定 TTL(Time to Live)等於 n ，向 hop 數小於 n 之範圍內的節點廣播出去，也因此可能會造成系統之負擔（訊息洪流：Message Flooding），且可能造成查詢沒有得到回應之情況。若改為本系統架構，由於所發出之查詢訊息將會先行到達 Super-Peer Network 上，再加上本系統架構中之「查詢過之關鍵字紀錄機制」，因此系統中所有比較熱門的分享檔案，將會更加快速的被使用者所搜尋到，系統訊息不但不再盲目的送出，而且較熱門之檔案可以被快速地搜尋到之優點相當符合點對點網路架構之精神。另外在 KRDB 同步訊息之發送也因為「雙環狀」架構而使得訊息傳遞具有方向性，如此一來，訊息將不會和之前一樣進行盲目式的廣播(Blind Broadcast)。因此訊息發送較前者具有方向性。

- KRDB 同步：

原本之架構由於系統中每個節點都擁有自己的 KRDB，而在進行 KRDB 同步之時，是靠著廣播之方式傳給鄰近的節點，尋找最佳的同步對象，如此一來，系統便無法進行整體的 KRDB 同步；而本系統架構，由於加入了 Super-Peer Network 的概念，因此在進行 KRDB 同步的時候，可以利用 SP 記載之左右鄰居節點進行同步，如此則可以讓系統進行整體的 KRDB 同步。

- KRDB 儲存空間：

KRDB 架構由於在系統中，每一節點都需要有一個自己的 KRDB 來記錄本節點上之 KR，因此會造成系統資源浪費。而本論文所提出之架構由於加入了 Super-Peer 網路的概念，因此每個分群共用一個 KRDB，而此 KRDB 由各個分群之 Super Peer 保管，備份則由候選點來做，如此一來，可以大大降低系統所有節點之儲存空間且可以達到系統 KRDB 共用的目標。



第五章 結論及未來研究方向

本章節將會對於本論文提出簡單的結論，說明本論文的貢獻，並對於未來的研究方向提出一些建議。

5.1 結論

本論文主要貢獻為建構一個應用於點對點網路的語義關鍵字搜尋系統架構。語義關鍵字搜尋在集中式架構中，由於可以將語義資料庫放在中央伺服器，因此可以輕鬆的完成；而在點對點網路中，由於節點之間並沒有一個中央伺服器可以存取，因此本論文主要的目的就是要讓語義關鍵字搜尋在點對點網路中能夠實行且效果能趨於集中式架構之效果。

本論文在第一章說明了本論文的研究動機，並簡介研究主題；在第二章裡，收集並且閱讀一些相關文獻以及書本文章，瞭解到點對點的網路架構之發展沿革、點對點網路架構之分類、Super Peer 網路之用途、語義關鍵字搜尋之發展、以及過去學者對於語義關鍵字搜尋在點對點網路[11]之研究。分析過去學者所提出之方法後發現其方法仍存在著許多尚待解決之問題。第三章為本論文研究之主體，說明了本論文所提出的系統架構及其系統運作流程，包括：「分群準則」、「以 Super Peer 為基礎之語義關鍵字搜尋機制」、「查詢過之關鍵字之快取機制」、以及因應 Super Peer Network 概念加入而改良的「KRDB 同步、評估與更新」之方法。

在第四章中，模擬實作本架構之相關機制，並進行成果分析以證明本論文所提出之架構可以在點對點網路中達到和以往集中式語義資料庫相同的效果且確實可以改善第三章中所提出之問題。

5.2 未來研究方向

本論文主要應用在點對點網路架構之檔案分享系統，因此在安全上的規範，顯得格外重要。例如：在本論文之架構中 Super Peer 要如何因應且防範其他節點對於本身的網路攻擊也是一個重要的問題。另外，由於本論文架構中之 Super Peer 網路是以「雙環狀」之方式架構，因此在傳遞訊息上可能沒有直接傳送給目的節點快，但是由於在點對點環境中無法在某節點紀錄系統中所有節點之資訊（像是路由表），且基於第三章之分析比較，「雙環狀」之架構對於本系統架構來說，仍是最好的選擇。



附錄

附錄(A)

4.4 節之 Metadata 正確率公式分析

Metadata 正確率(Pi)代表此方法架構下所能搜尋到的目標檔案數量與傳統點對點檔案搜尋系統所能搜尋到的目標檔案數量之比例。 Pi 越高則代表此方法架構所提供之 metadata 資訊越充足，這也代表了，本方法架構將可趨近於集中式語義資料庫所達到的效果。

4.4 節之之公式(4-1) (4-2)如下所示：

$$Pi(KRDB) = 1 - \frac{Ni^{Org}}{Ni^{KRDB}} \quad (4-1)$$

$$Pi(SP - KRDB) = 1 - \frac{Ni^{Org}}{Ni^{SP-KRDB}} \quad (4-2)$$

此公式是參考 Kiyohide Nakauchi 等人提出之 KRDB 架構論文[11]的評估方法，原本公式為

$$Pi = \frac{Ni^{Org}}{Ni^{Sem}}$$

Ni^{Org} ：傳統點對點檔案搜尋系統所能搜尋到之目標檔案數量

Ni^{Sem} ：語義關鍵字 KRDB 方法架構所能搜尋到之目標檔案數量

此公式中，以 Ni^{Org} 除以 Ni^{Sem} 之值越高代表所能搜尋到的目標檔案越少；反之，則搜尋到的檔案越多。 Pi 代表 Metadata 正確率，因此此公式得出之值可能會讓人產生誤會（ Pi 值高，卻代表搜尋到的檔案數量少，系統提供之 metadata 不夠充足）。因此本論文將此公式改為 4.4 節之之公式(4-1) (4-2)。

以下為 4.4 節實驗中，說明檔案搜尋結果 metadata 正確率 Pi 之計算範例，如圖 29 所示：

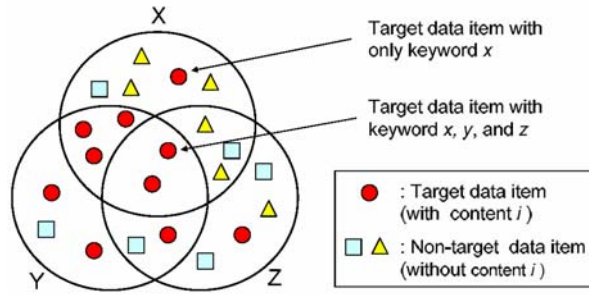


圖 30 搜尋結果之範例

圖 30 為一個檔案搜尋結果之範例。此範例將會說明 4.4 節之公式(4-1)(4-2)要如何計算，並利用其作為本論文成果分析之根據。實驗中，將欲找到之檔案稱為「目標檔案」，為了要找到這些目標檔案，使用者必須輸入一些關鍵字來做搜尋。本次實驗過程中，會先行給定一個特定的關鍵字 x(以下簡稱 Keyword x)，而檔案內容含有 Keyword x 的目標檔案則定義為集合 X(以下簡稱 Set X)。接著選定另外兩個和 x 有相關之關鍵字：y 和 z(以下簡稱 Keyword y, Keyword z)。在圖 30 中，圓點代表搜尋者鎖定的目標檔案(代表這些檔案為搜尋者所欲搜尋的檔案)，而在此實驗中，目標檔案可分為以下三種：

- 只有和 Keyword x, y, z 個別相關：

$$Xi \cap \overline{Yi} \cap \overline{Zi}, \overline{Xi} \cap Yi \cap \overline{Zi}, \overline{Xi} \cap \overline{Yi} \cap Zi$$

- 分別和 Keyword x, y, z 其中兩者相關：

$$Xi \cap Yi, Yi \cap Zi, Xi \cap Zi$$

- 和 Keyword x, y, z 均相關： $Xi \cap Yi \cap Zi$

根據上述範例可將 4.4 節之公式(4-1)(4-2)可將 P_i 轉換成以下公式：

$$P_i = 1 - \frac{Ni(Xi \cap Yi \cap Zi)}{Ni(Xi \cup Yi \cup Zi)}$$

$Ni(Xi \cap Yi \cap Zi)$ ：在傳統點對點檔案搜尋系統中，若要尋找和 x, y, z 均相關的檔案時，所能搜尋到之目標檔案數量

$Ni(Xi \cup Yi \cup Zi)$ ：系統中所有包含在 X、Y、以及 Z 集合中所有語義關鍵字搜尋系統可搜尋到的目標檔案數量

而 $Ni(Xi \cup Yi \cup Zi)$ 又可分解成以下公式：

$$\begin{aligned}
Ni(Xi \cup Yi \cup Zi) = & \\
& Ni(Xi \cap Yi \cap Zi) + Ni(Xi \cap Yi) + Ni(Yi \cap Zi) + Ni(Xi \cap Zi) + \\
& Ni(Xi \cap \overline{Yi} \cap \overline{Zi}) + Ni(\overline{Xi} \cap Yi \cap \overline{Zi}) + Ni(\overline{Xi} \cap \overline{Yi} \cap Zi)
\end{aligned}$$

如此一來，將可以計算出 Pi 的值。例如，在圖 30 中，各項之值如下所示：

$$Ni(Xi \cap Yi \cap Zi) = 2, Ni(Xi \cap Yi) = 3, Ni(Yi \cap Zi) = 1,$$

$$Ni(Xi \cap Zi) = 0, Ni(Xi \cap \overline{Yi} \cap \overline{Zi}) = 1, Ni(\overline{Xi} \cap Yi \cap \overline{Zi}) = 2,$$

$$Ni(\overline{Xi} \cap \overline{Yi} \cap Zi) = 1$$

因此

$$Pi = 1 - \frac{Ni(Xi \cap Yi \cap Zi)}{Ni(Xi \cup Yi \cup Zi)} = 1 - \frac{2}{2 + 3 + 1 + 0 + 1 + 2 + 1} = 0.8$$

接著說明 $Pi(x), Pi(y), Pi(z)$ ：

$Pi(x)$ ：僅以關鍵字 x 在傳統點對點網路中尋找所得之數量與在語義關鍵字

搜尋系統中所得之數量之比較

$Pi(y)$ ：僅以關鍵字 y 在傳統點對點網路中尋找所得之數量與在語義關鍵字

搜尋系統中所得之數量之比較

$Pi(z)$ ：僅以關鍵字 z 在傳統點對點網路中尋找所得之數量與在語義關鍵字

搜尋系統中所得之數量之比較

計算本例中 $Pi(x), Pi(y), Pi(z)$ 的值：

$$Pi(x) = 1 - \frac{Ni(Xi)}{Ni(Xi \cup Yi \cup Zi)} = 1 - \frac{6}{2 + 3 + 1 + 0 + 1 + 2 + 1} = 0.4$$

$$Pi(y) = 1 - \frac{Ni(Yi)}{Ni(Xi \cup Yi \cup Zi)} = 1 - \frac{8}{2 + 3 + 1 + 0 + 1 + 2 + 1} = 0.2$$

$$Pi(z) = 1 - \frac{Ni(Zi)}{Ni(Xi \cup Yi \cup Zi)} = 1 - \frac{4}{2 + 3 + 1 + 0 + 1 + 2 + 1} = 0.6$$

以上為 4.4 節中 Metadata 公式之計算方式，若能證明本論文架構之 Pi 比 KRDB 架構之 Pi 大，則代表本論文的確可以達到較好之語義關鍵字搜尋之結果。

參考文獻

- [1] Peer-to-Peer Working Group Committees, <http://peer-to-peerwg.org/>
- [2] Gnutella website, <http://gnutella.wego.com>
- [3] Kara, <http://www.kazaa.com/>
- [4] SETI@home, <http://setiathome.ssl.berkeley.edu/>
- [5] JXTA Project, <http://www.jxta.org>
- [6] Napster, <http://www.napster.com>
- [7] Y. Beverly, G. M. Hector, "Designing a super-peer network," Technical report, Stanford University, 2002.
- [8] Alexander Löser, Felix Naumann, Wolf Siberski, Wolfgang Nejd, Uwe Thaden, "Semantic Overlay Clusters within Super-Peer Networks", DBIS P2P, 2003.
- [9] Dejan S. Milojevic, Vana Kalogeraki, Rajan Lukose, Kiran Nagaraja¹, Jim Pruyne, Bruno Richard, Sami Rollins², Zhichen Xu, Peer-to-Peer Computing, HP Tech. Report, March 8th, 2002.
- [10] Clarke, O Sandberg, B Wiley, TW Hong, "Freenet: A Distributed Anonymous Information Storage and Retrieval System", Workshop on Design Issues in Anonymity and Unobservability, 2000
- [11] Kiyohide Nakauchi, Hiroyuki Morikawa, and Tomonori Aoyama, "Design and Implementation of a Semantic Peer-to-Peer Network", IEEE High Speed Networks and Multimedia Communication, 2004
- [12] Kiyohide Nakauchi, Hiroyuki Morikawa, and Tomonori Aoyama, "Peer-to-Peer Keyword Search Using Keyword Relationship", 2003.
- [13] Stoica, R Morris, D Karger, MF Kaashoek, "Chord: A scalable peer-to-peer lookup service for internet applications", SIGCOMM, 2001.
- [14] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network," in *SIGCOMM 2001*, San Diego, CA.
- [15] G. Kwon, K.D. Ryu, "An Efficient Peer-to-Peer File Sharing Exploiting Hierarchy and Asymmetry," Applications and the Internet, 2003. Proceedings. 2003 Symposium on, 27-31 Jan. 2003, pp. 226-233.
- [16] Salman A. Baset and Henning Schulzrinne, "An Analysis of the Skype Peer-to-Peer Internet Telephony," September 15, 2004.
- [17] AIT Rowstron, P Druschel, "Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems", Middleware, 2001.
- [18] B. Y. Zhao, J. D. Kubiatowicz, and A. D. Joseph, "Tapestry: An infrastructure for

fault-tolerant wide-area location and routing,” Tech. Rep. UCB//CSD-01-1141, Apr. 2000

[19] Maymounkov, P., Mazieres, P.: Kademlia: a Peer-to-Peer Information System based on XOR Metric. First International Workshop on Peer-to-Peer Systems (IPTPS)(2002).

[20] Aberer, K.: P-Grid: A Self-organizing Access Structure for **P2P** Information Systems. Sixth International Conference on Cooperative Information Systems (CoopIS 2001), Trento, Italy (2001)

[21] ID3 tag , <http://www.id3.org/>

[22] 線上音樂發展與趨勢分析,經濟部工業局,資訊工業策進會,2003年9月

