具位置知覺能力之混合式 P2P 興趣分群系統

曾黎明　　胡鈞証　　陳彥儒
國立中央大學資訊工程研究所
tsenglm@csie.ncu.edu.tw , horsy@dslab.csie.ncu.edu.tw, oldbig@dslab.csie.ncu.edu.tw

摘要

同儕網路 (Peer-to-Peer；P2P) 近幾年來被大量的應用在網路服務上，而一個同儕系統如果具有位置知覺能力的話將會大幅增加系統效能。所以本篇論文提出利用層級架構 (hierarchical structure)來達成位置知覺並且將使用者依興趣分群的混合式 P2P系統。在本系統中，使用者會依照位置區域和興趣分群並且利用超立方體結構來替使用者定址並利用選舉出的超級節點 (super node) 來替其底下普通節點 (peer node)來進行跨區域與跨興趣的詢問。我們還為系統運作設計了三種通訊協定，分別是節點加入、離開與搜尋通訊協定。最後我們利用 NS2 來模擬系統，並分別以點對點延遲與詢問所經過的 hop 數來評估本系統的效能，模擬的結果顯示出在依時間順序加入系統的節點位置是均勻分散且網路節點夠多的情況之下本篇論文所提出的方法會大大的提升系統效能。

關鍵詞：混合式 P2P 系統、位置知覺、興趣分群、超立方體結構。

1. 前言

同儕網路 (Peer-to-Peer；P2P) 近幾年來被大量的應用在網路服務上，由本來的單純的檔案交換，演變成其他商業應用。P2P 並沒有一個統一的定義，但是 P2P 大致都是符合下面的模式，每個主機本身既是 Client 端，抑是 Server 端，而且每個主機彼此可以對等的互相溝通聯絡。

目前的同儕網路可依架構分類成四大類，分別是集中式系統、分散式非架構化系統、分散式架構化系統與混合式系統。集中式系統以 Napster [13] 為代表，Napster 是最早出現的純檔案分享同儕網路，在這種網路架構中，最大的優點就是維護簡單，而且搜尋效率高，然而在這種系統中會遇到單點失敗問題。而分散式非架構化系統就不是利用集中伺服器來管理資料目錄，而是在覆蓋網路上透過在 TTL 來泛洪詢問訊息，這種架構的好處就是因為沒有集中伺服器，所以系統不會遭遇到單點失敗問題，在使用者數量多的情況下也能運作，最重要的是它在高動態網路下還是可以正常運作，可是因為分散式非架構化系統主要是用泛洪的方法來發布訊息，所以在網路規模不斷變大的情況之下，網路流量將會激增，而且在 TTL 的時間內會因為網路使用者太多而只能搜尋到某一部份網路導致搜尋效率低落，所以有許多論文都致力於如何增進分散式非架構化系統的效能 [14][15][16]。

由於分散式非架構化系統搜尋的效率低落，所以大部分的論文都集中在如何改善這種分散式非架構化系統的搜尋效率，所以就衍生出高搜尋效率的分散式架構化系統，用來解決搜尋效率低落的問題，像是 Chord [17]、Pastry [18]等。分散式架構化系統，都是使用 DHT (Dynamic Hash Table) 來將關鍵字映射到某個節點，然後再根據各方法的路由策略來快速搜尋到資料。分散式架構化系統最大的問題就是節點頻繁的加入離開系統會為 DHT 的維護帶來極大的負擔。

有鑑於以上的系統都各有優缺點，所以就有混合式系統的出現，如 Kazza [19]，這種系統吸收了集中式系統與分散式非架構化系統的優點，在這種系統中會選擇擁有較高系統資源並且可靠性較高的的節點，來當作超級節點 (super node)，不同超級節點下的底層節點利用超級節點來轉發訊息，形成一種層級架構 (hierarchical structure)。

在 P2P 系統中，位置知覺是一個很重要的議題，因此本篇論文提出一種利用層次架構來達成位置知覺並且將使用者依興趣分群的混合式 P2P 系統，。在本系統中，使用者會依照位置區域和興趣分群並且利用超立方體結構 (Hypercube structure) 來替使用者定址。整個系統從位置區域的觀點來看，系統將依實體位置相近的使用者分在同一個區域之內，從興趣的觀點來看，系統將把使用者邏輯上的分成若干個興趣叢集 (Interest Cluster)，然後經由選舉出的超級節點 (super node) 來替其底下普通節點 (peer node)來進行跨區域與跨興趣的詢問，。以下的論文架構如下：第二章是一些相關研究，會描述關於如何利用興趣叢集和位置知覺能力來改善 P2P 系統效能的方法；第三章說明本系統的設計與方法；第四章是系統的實驗模擬；第五章為本篇論文的結論。

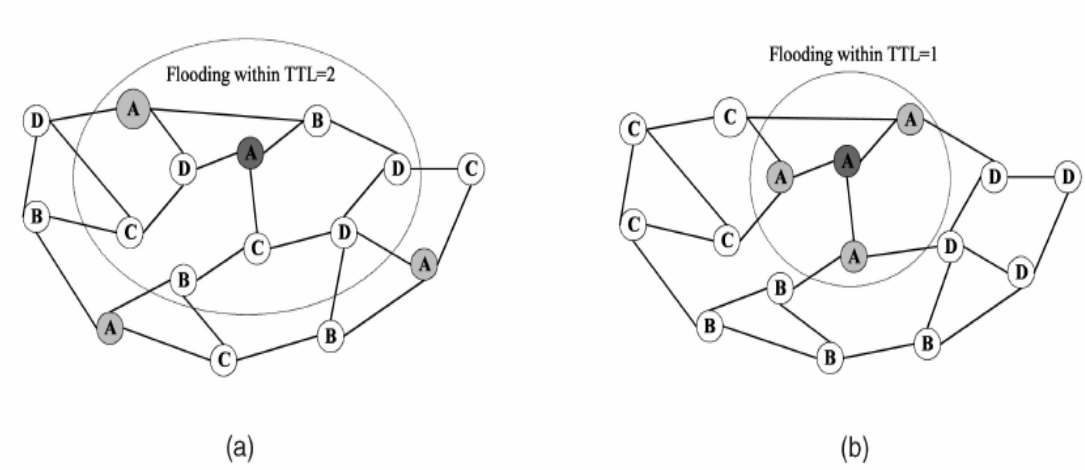


圖 1 將使用者依興趣分群示意圖（摘自［6］）

2. 相關研究

2.1 興趣叢集

將使用者依興趣分類成叢集主要的目的就是要改善分散式非架構化系統中詢問花費過高與搜尋效率低落的問題。如圖 1 所示。在圖 1(a)中的網路是由 16 個使用者所組成的，而這 16 個使用者又可依興趣分成四個主題分別是 *A*、*B*、*C*、*D*，根據圖 1(a)中的網路拓撲 (topology)，深灰色的使用者如果對在同樣主題 *A* 下的其他使用者做出詢問的話，在 TTL (Time To Live) 等於 2 的情況下，將只能詢問到一個使用者，而且在 TTL 等於 2 的時間內還會詢問到其他不相關的三個主題。如果我們將所有的使用者依四個主題邏輯上分成叢集的話，如圖 1(b)，深灰色的使用者在 TTL 等於 1 的時間內就可以詢問到所有在主題 *A* 下的使用者，而且也不會詢問到其他不相關的主題。如此以來由於詢問將只發生在同樣的主題下，不僅使得詢問的花費大量的減少，而且如果只針對相關的主題去做搜尋的話，搜尋命中率 (hit rate) 也相對大大的增加 [6]。

覆蓋網路 (overlay network) 使以上的構想得以實現 (implement)，也因為覆蓋網路的特性讓興趣叢集的實現方式多元化。興趣叢集的實現方式大致可以分做兩類，一類是在原本的承載網路拓撲中為興趣相同的使用者直接建立捷徑 [3][4][5]，另一類則是在直接在覆蓋網路拓撲上建立興趣叢集 [2][6][7]。

2.2 位置知覺

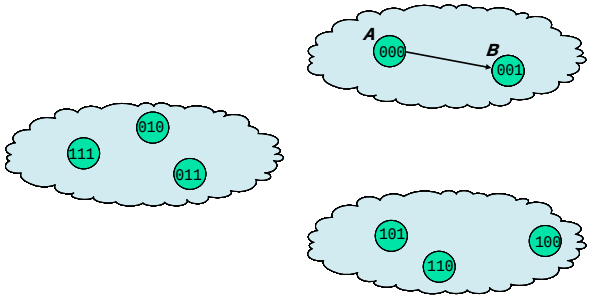
然而在覆蓋網路被大量應用之下產生了一些問題，因為建立覆蓋網路拓撲時並不需要知道承載網路的拓撲資訊 (information)，所以承載網路拓撲將會對覆蓋網路的效能產生極大的影響。其中最常被討論的問題有兩個，分別是錯誤配對問題與訊息重複問題，關於如何利用承載網路的位置資訊 (locality information) 來改進覆蓋網路的效能問題已經被討論了很多年，為了要解決上面所提出的兩個問題，許多研究具有位置知覺能力的網路系統論文被提出來，在 [8][9]中的方法都是透過探索 (probing) 鄰居後，利用所獲得的資訊來重建覆蓋網路拓撲。除此之外也有一些論文致力在分散式架構化網路 (Decentralized/structured system) 的位置知覺上，如 [10][11]，原本 Chord ring 上的一個節點是代表一個使用者，然而如果一個節點只代表一個使用者的話，Chord 便難以具有位置知覺能力，所以這兩篇論文都提出一個節點代表一個地理位置鄰近區域的做法，在 [10]中每個區域選出一個代理者 CRN (Chord Replica Node) 來代表區域中其他的使用者 ORN (Ordinary replica node)。而在 [11]中則是鄰近區域裡的所有使用者都擁有同樣的 Chord ring 節點 ID，並且一起管理 Chord ring 上一個節點的資料。

3. 系統設計

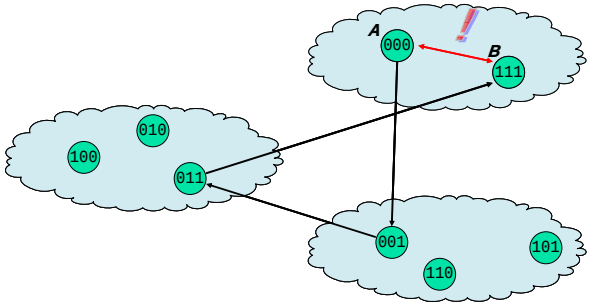
3.1 動機與目的

論文［6］利用興趣叢集來改善分散式非架構化系統效能低落的缺點，並且利用超立方體的架構來提升搜尋效率並解決訊息重複問題。然而如果在隨著時間加入系統的節點實體位置分布分散的情況下，錯誤配對的問題將會發生，如圖 2 所示。假設隨著時間節點加入系統並且定址的順序是 000、001、010、011、100、101、110、111，在圖 2(a)中，節點 *A* 所要搜尋的檔案如果是由節點 *B* 所管理的話，那麼 *A* 只要經過一個 hop 就可以詢問到 *B*，而在圖 2(b)中，由於 *A* 是第一個加入系統的節點，*B* 卻是最後一個，所以 *A* 雖然只經過兩個 hops 就詢問到 *B*，但是實際上卻是經過兩個自主系統才能到達點 *B*。所以我們可以發現詢問所走的路徑會遠大於兩點間實體上最短的距離，另外一方面也可以發

現當節點隨著加入系統順序上的不同將會直接影響到使用超立方體架構來定址的系統其效能。因此本論文提出具位置知覺能力之混合式 P2P 興趣分群系統 (A Locality Awared Hybrid P2P system with Interest Grouping，簡稱 LaHiG)，其目的就是希望在引用超立方體的架構下還能夠使節點加入順序對系統效能的影響程度降到最低。



(a) 較有效率的超立方體定址



(b) 較無效率的超立方體定址

圖 2 超立方架構之錯誤配對問題

3.2 系統組成

假設整個系統有 m 個位置區域 (locality region)，n 個興趣叢集 (interest cluster)，我們用下面的符號來詳細說明系統架構。

- $R\,i, 0 \leqq i \leqq m-1$：位置區域。我們將地理位置靠近的節點 (以下通稱 peer node) 組成一個位置區域，有很多方法可以讓我們得到 peer node 地理位置的資訊，例如可以利用 *IP* 位址的 prefix 來判斷，越靠近的點其 prefix [*IP address*] 就越相似 [12]，或是透過 ping 來判斷。
- $I\,j, 0 \leqq j \leqq n-1$：興趣叢集。本系統需要為興趣叢集建立索引 (index)，索引的位元數為 $\lceil \lg n \rceil$，然後每個不同的索引值就對應到一個興趣叢集，舉例來說，要是系統有四個興趣叢集，那麼這四個興趣叢集的索引依序會是 00、01、10、11，而這興趣索引將會用在超級節點 (以下通稱 Super node) 的定址上。
- $R\,I\,i\,j, 0 \leqq i \leqq m-1, 0 \leqq j \leqq n-1$：我們可以再將所有的節點依照位置區域與興趣叢集分成 $m \times n$ 個集合。
- $X\,i\,j = \{ X\,k \mid X\,k \in R\,I\,i\,j \}$：代表在 ith 位置區域且在 jth 興趣叢集中所有 node 的集合，這些 node 將使用第一層超立方體 (layer-one hypercube) 來定址並用興趣內連結 (intra-cluster link) 來建立連線，每個 $X\,i\,j$ 用來定址的維度至少需要 $\lceil \lg |Xij| \rceil$，而且每個 $X\,i\,j$ 都會有屬於自己的超立方體位址而不會互相共用，需要舉例來說，$X_{1\,2}$ 如果有 1024 個 node，那他將至少需要用維度 10 的超立方體來定址，而另一個集合 X_{23} 如果只擁有 128 個 node 的話，那麼它可以只用維度 7 的超立方體來定址。
- $S\,i\,j, S\,i\,j \in X\,i\,j$：代表由在 i 位置區域且在 j 興趣叢集中的所有 peer node 所選出的 super node。因為 super node 必須幫 peer node 做出跨興趣與跨區域的查詢，所以它必須擁有較高的頻寬與資料處理速度，而且不能任意的離開與加入系統，所以 super node 需由擁有較高資源且可信度較高的 node 擔任。其中 $S\,i_{\,1}$、$S\,i_{\,2} \cdots S\,i\,n$ 是完全連結網路拓樸 (fully connected mesh topoloty)，而且在這完全連結網路拓樸中所有的連線都是最短的，也就是說每個興趣叢集裡的 super node 都會去尋找其他興趣叢集中離他最近的 super node 當作鄰居。而 $S_{\,1}\,j$、$S_{\,2}\,j \cdots S\,m\,j$ 是用第二層超立方體來定址。

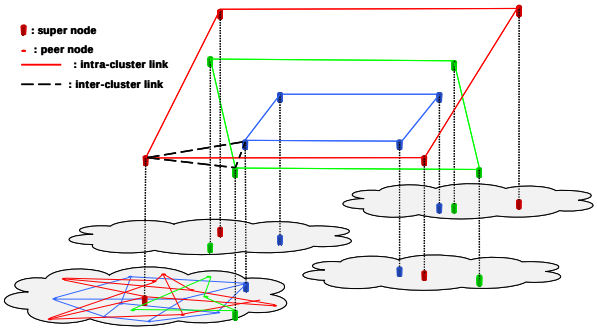


圖 3 完整的系統架構

圖 3 是一個完整的系統例子，在這個例子中 node 顏色代表興趣叢集 (紅、綠、藍)，我們可以看到同樣顏色下的 peer node 與 super node 分別建構成第一層超立方體與第二層超立方體，而且在同一個區域中的 super node 形成完全連結網路拓樸。

我們分別為雙層超立方體設計了兩種廣播程序，分別是 *Layer-1_Broadcast*、*Layer-2_Broadcast*，這兩個廣播程序也是整個系統最主要的部分，後面的通訊協定都是基於這兩個程序運作的。圖 5 是 *Layer-1_Broadcast* 程序，用來負責 $X\,i\,j$ 內的訊息廣播。我們可以看到 *Layer-1_Broadcast* 有四個參數，*dim* 是 $X\,i\,j$ 的維度，*msg* 是廣播的訊息，*NA* 是正要執行程序的 node 位址，當一個 node 收到另一個 node

Layer-one broadcasting procedure

```
Layer-1_Broadcast(dim,msg,NA,count)
    for (i=count to dim-1){
        if (i=0)
        sent query msg to super node;
        DA = (NA) XOR 2^i ;
        sent (dim,msg,DA,i+1) ;
    }
```

圖 5 Layer-1_Broadcast

Layer-two broadcasting procedure

```
Layer-2_Broadcast(dim,msg,SNA,count)
    for (i=count to dim-(length(index[SNA])+1)){
        SDA = (SNA) XOR 2^i ;
        sent (dim,msg,DNA,i+1) ;
    }
```

圖 6 Layer-2_Broadcast

由程序第五行所送出的訊息，它將立刻執行 *Layer-1_Broadcast*，值得注意的是一開始送出訊息的 node 也會送出訊息給它所屬的 super node。而圖 6 是 *Layer-2_Broadcast* 程序，*Layer-2_Broadcast* 則是負責 S_1j、S_2j…Smj 間的廣播，其運作的方式與 *Layer-1_Broadcast* 大致相同，差別只在於只需對除了索引外的位址做廣播。值得注意的是 super node 也會同時扮演 peer node 的角色，所以 super node 除了會執行 *Layer-2_Broadcast* 程序外，也會執行 *Layer-1_Broadcast* 程序。

3.2 通訊協定

■ 加入通訊協定

A 是想要加入系統的節點，而 I_A 是其想要加入的興趣叢集，加入通訊協定包含了下面四個階段：

1. 首先我們必須利用使用者清單找到一個已經在系統中的節點。

2. 利用第一階段中求得的 node 來找到所要加入的興趣叢集。假設第一階段所找到的 node 叫 *B*，*A* 會先送出加入請求 (Join Request) *JRQ* 訊息給 *B*，*B* 會將 *JRQ* 訊息轉送給 *B* 所屬的 super node *S*，*S* 會檢查 I_A 是否等於 I_S，如果相等的話，*S* 就是第二階段所要找的 node；反之如果不相等的話，，*S* 會利用它路由表中的鄰居欄位中各位址的索引值來找到興趣叢集是 I_A 的 super node *S'*，*S'* 就是本階段所要找的點。

3. 利用第二階段找到的 super node 來找到最近的區域位置。*A* 會透過由第二階段所找到的 super node *C* 其路由表的鄰居，反覆詢問來找到離 *A* 最近的 super node *D*，然後 *C* 會送出 *JRQ* 訊息給 *D*。

4. 從第三階段求得的位置區域中找到還有其他第二位址的 peer node。假設第三階段所找到的 super node 是 *D*，*D* 會開始對由它為樹根的廣播樹對底下的 peer node 作深度優先搜尋 (Depth First Search)，直到找到一個還擁有第二位址的 peer node *E*，*E* 就是本階段所要求的 node，然後同樣的 *D* 會送出 *JRQ* 訊息給 *E*。

5. 分配位址並更新路由表。第四階段所找到的 node *E*，將會利用前面所提過的位址分配方式來替 *A* 定址，並且會送出加入確認 (Join Confirm) *JCF* 給 *A*，
並且更新 *A* 與 *E* 的路由表。

■ 離開通訊協定

假設 *A* 是欲離開系統的節點，離開通訊協定只有兩個階段：

1. 從欲離開系統節點路由表的鄰居欄位中找到擁有最小鄰居位址的節點 *B*。

2. 對 *B* 送出離開請求 (Leave Request) *LRQ*，並將 *A* 擁有的位址交給 *B*，*B* 在收到 *A* 送出的訊息後便開始更新路由表，並且發出一個通知訊息給所有 *B* 的鄰居，告知他們所有對應到 *A* 真正 IP 的位址都改成對應到 *B* 的 IP，最後 *B* 會送出離開確認 (Leave Confirm) *LCF* 訊息給 *A*，然後 *A* 就會離開系統完成離開通訊協定。

■ 搜尋通訊協定

假設 *A* 想要對興趣叢集 I_A 發出詢問，*A* 會透過三個階段來取得詢問結果。

1. *A* 要先找到正確的興趣叢集 I_A。*A* 會先檢查本身是否處在 I_A 之下。

2. 如果 *A* 如果就在所要的興趣叢集之下，那麼 *A* 就是本階段要找的點；反之如果不是處在興趣叢集 I_A 下，*A* 就會直接去詢問它所屬的 super node *S*，*S* 再經由它的路由表找到興趣叢集 I_A 中離 *S* 最近的 super node *S'*，*S'*就是本階段要找的點。

3. 利用第一階段所找到的點做廣播。如果第一階段找到的點是 peer node 的話，它會在執行完 *Layer-1_Broadcast* 的同時送出訊息給它所屬的 super node，然後 super node 在收到訊息後會馬上執行 *Layer-2_Broadcast*；如果第一階段找到的點是 super node 的話，它將會同時執行 *Layer-1_Broadcast* 與 *Layer-2_Broadcast*。

4. 每一個收到詢問的節點都會查詢自己的資料庫看是否有 *A* 所詢問的項目，如果有的話便將結果回傳給 *A*。

4. 效能評估

4.1 實驗環境

下來我們利用 NS2 (Network Simulator version 2)來模擬系統環境，並且把 PeerCluster 當作效能評估比較的對象。以下是在實驗中所用的設定：

- 興趣叢集數量 4 個。
- 位置區域數量 8 個。
- 節點的連線限制 (link constrain) 設定為 4 個。
- 詢問興趣叢集以 80/20 分布，也就是有 80%的詢問是發生在節點主要的興趣叢集之下。
- 有 20%的節點擁有詢問的結果。

我們將由兩方面來評估系統效能，一個是點對點延遲 (End-to-End delay)，另一個是詢問所花費的 hop 數。因為我們希望用點對點延遲來量測詢問的路由路徑距離，所以在實驗中我們將所有的頻寬都限制在 100MB，然後依節點間的距離來設定彼此間的連線延遲 (link delay)，同時為了減少瓶頸頻寬 (bottleneck bandwidth) 對於點對點延遲的影響，所以同一時間內只會有一條資料流 (data flow)。然後我們在模擬設定上，依時間順序加入系統的節點位置是均勻分散的，也就是鄰近的節點不會大量的在短時間內一起加入系統。

4.2 實驗結果與分析

首先我們先來看點對點延遲的模擬結果，如圖 7 所示。我們分別對網路節點數量是 32、64、128、256、512、1024 的情況下做模擬，在圖 7 中可以發現在網路節點數量夠多的時候，LaHiG 在點對點延遲遠低於 PeerCluster，在網路節點數超過 256 個的情況下，LaHiG 的點對點延遲比 PeerCluster 快了 6.63 倍，也就是意味著在 PeerCluster 架構下的路由路徑將會比 LaHiG 多上 6.63 倍。然而在網路節點很少的時候 (256 個節點以下)，因為節點數已經不足夠使得每一個位置區域裡的節點擁有詢問的結果，所以詢問出現跨區域的情況增加，導致節點數在 256 個以下的時候，LaHiG 的點對點延遲快速提升，但因為跟 PeerCluster 比，LaHiG 還是具有程度上的位置知覺能力，所以點對點延遲還是相對的比較低一點。圖 8 則是每個節點詢問的點對點延遲在不同網路節點數下的分布情況，我們可以發現在 PeerCluster 下，因為不具位置知覺能力，所以每個節點詢問的點對點延遲分布區域很大，在網路節點數夠多的情況下點對點延遲最高可到達 3.594600 秒，最低是 0.111084 秒，而在 LaHiG 下因為具有位置知覺能力，所以網路節點數夠多的情況下，點對點延遲分布會集中在一起，代表著大多數的點都會在鄰近的區域中詢問。圖 9 是在不同節點數下兩種系統詢問所經過的 hop 數比較。在圖 9 中可以發現在網路點數多 (超過 256 個) 的情況之下是跟 PeerCluster 差不多的，然而在網路節點數量不夠多

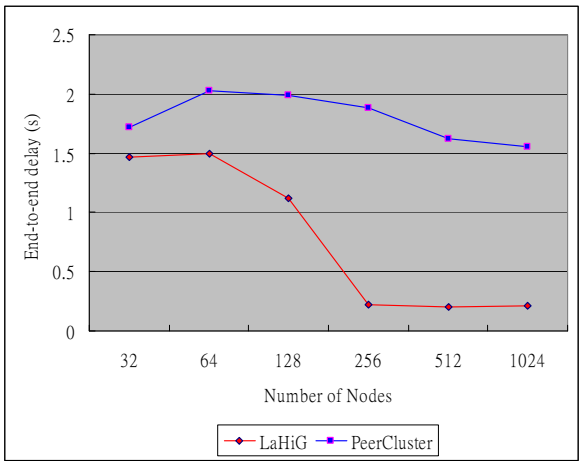


圖 7 在不同節點數下的點對點延遲比較

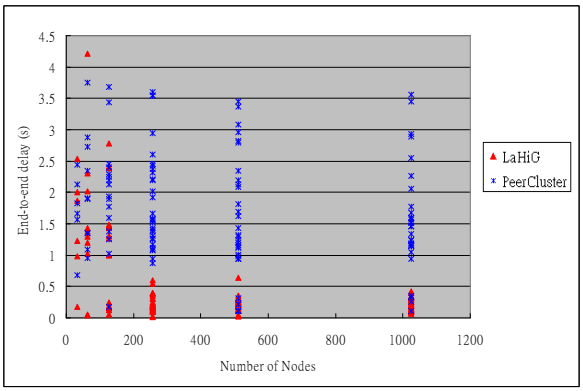


圖 8 點對點延遲分布圖

的情況之下 (<256)，因為在 LaHiG 下，跨區域詢問的情況增加，所以 peer node 會一直透過 super node 來進行跨區域詢問，比起在位置區域內的詢問至少就多了一個 hop 來路由到 super node，導致詢問的 hop 數大量增加，而 PeerCluster 因為不需要多花費 hop 在 super node 上，所以 hop 數變化相差不大。所以經由以上兩個實驗結果，證明了在依時間順序加入系統的節點位置是均勻分散且網路節點夠多的情況之下本篇論文所提出的方法會大大的提升系統效能。

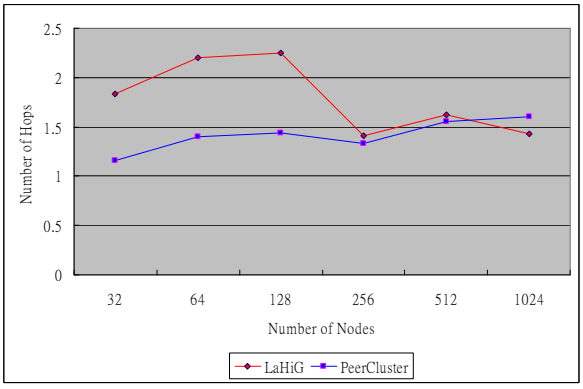


圖 9 在不同節點數下詢問所經過的 hop 數比較

5. 結論

本篇論文提出具位置知覺能力之混合式 P2P 興趣分群系統來改善利用超立方體結構來定址會產生錯誤配對的問題，具位置知覺能力之混合式 P2P 興趣分群系統採用雙層超立方體架構，並且將所有的使用者依照興趣叢集與位置區域來分類，並且利用 super node 來幫助 peer node 作跨興趣與跨區域的詢問。同時我們也在論文中提出了雙層超立方架構的廣播方法，並且也說明節點加入通訊協定、節點離開通訊協定、節點搜尋通訊協定，利用這三個通訊協定便可讓具有位置知覺能力之混合式 P2P 分群系統成功運作。

在最後我們做了與 PeerCluster 比較的模擬，模擬結果說明了在依時間順序加入系統的節點位置是均勻分散且網路節點夠多的情況之下本篇論文所提出的方法比起 PeerCluster，點對點延遲將會降低 6.63 倍，證明了在超立方結構下結合位置知覺可以大大的提升系統效能。

參考文獻

[1] Busnel, Y., Kermarrec, A.-M., “PROXSEM: Interest-Based Proximity Measure to Improve Search Efficiency in P2P Systems,” 4th European Conference on Universal Multiservice Networks, pp. 62-74, Feb. 2007.

[2] Kobayashi, H., Takizawa, H., Inaba, T., Takizawa, Y., “A Self-Organizing Overlay Network to Exploit the Locality of Interests for Effective Resource Discovery in P2P Systems,” IEEE Symposium on applications and the Internet, pp. 246-255, Feb. 2005.

[3] Wen-Tsuen Chen, Chi-Hong Chao, Jeng-Long Chiang,” An Interested-based Architecture for Peer-to-Peer Network Systems,” 20th IEEE international Conference on Advanced Information Networking and Applications, Volume 1, pp. 707 – 712, April 2006.

[4] Xi Tong, Dalu Zhang, Zhe Yang, ”Efficient Content Location Based On Interest-Cluster in Peer-to-Peer System,” IEEE international Conference on e-Business Engineering, pp. 324 – 331, Oct. 2005.

[5] Sripanidkulchai, K., Maggs, B., Zhang, H., “Efficient Content Location Using Interest-Based Locality in Peer-to-Peer Systems,” IEEE Infocom on Computer and Communications Societies, Volume 3, pp. 2166 – 2176, April 2003.

[6] Xin-Mao Huang, Cheng-Yue Chang, Ming-Syan Chen, “PeerCluster: A Cluster-Based Peer-to-Peer System,” IEEE Trans. on Parallel and Distributed Systems, Volume 17, Issue 10, pp. 1110 – 1123, Oct. 2006.

[7] Schlosser, M., Sintek, M., Decker, S., Nejdl, W., “A Scalable and Ontology-Based P2P Infrastructure for Semantic Web Services,” in IEEE Proc. of Second International Conference on Peer-to-peer computing, pp. 104 – 111, Sept. 2002.

[8] Yunhao Liu, Li Xiao, Xiaomei Liu, Ni, L.M., Xiaodong Zhang, “Location Awareness in Unstructured Peer-to-Peer Systems,” IEEE Trans. on Parallel and Distributed Systems, Volume 16, Issue 2, pp. 163 – 174, Feb 2005.

[9] Xiao, L., Yunhao Liu, Ni, L.M., “Improving Unstructured Peer-to-Peer Systems by Adaptive Connection Establishment,” IEEE Trans. on Computers, Volume 54, Issue 9, pp. 1091 – 1103, Sept. 2005.

[10] Zhenyu Li, Gaogang Xie, Zhongcheng Li, “Locality-Aware Consistency Maintenance for Heterogeneous P2P Systems,” in Proc. of IEEE International Parallel and Distrubuted Symosium, pp. 1-10, March 2007.

[11] Locher, T., Schmid, S., Wattenhofer, R., “eQuus: A Provably Robust and Locality-Aware Peer-to-Peer System,” 6th IEEE International Conference on Peer-to-peer Computing, pp. 3-11, Sept. 2006.

[12] Zhang, Guoqiang; Zhang, Guoqing, “Agent Selection And P2P Overlay Construction Using Global Locality Knowledge,” IEEE International Conference on Networking, Sensing and Control, pp. 519-524, April 2007.

[13] Napster Inc., http://free.napster.com, 2007.

[14] Kumar, A.; Xu, J.; Zegura, E.W.,”Efficient and Scalable Query Routing for Unstructured Peer-to-Peer Networks,” in Proc. of IEEE Infocom, Volume 2, pp. 13-17, March 2005.

[15] Qianbing Zheng, Xicheng Lu, Peidong Zhu, Wei Peng, “**An efficient random walks based approach to reducing file locating delay in unstructured P2P network**, “ IEEE Global Telecommunication Conf., Volume 2,pp. 5, Dec. 2005.

[16] Xiaomei Liu, Yunhao Liu, Li Xiao, “**Improving Query Response Delivery Quality in Peer-to-Peer Systems,” IEEE Trans. on Parallel and Distributed Systems,** Volume 17, Issue 11, pp. 1335 – 1347, Nov. 2006.

[17] I. Stoica, R. Morris, D. Karger, E Kaashoek and H. Balakrishnan.“Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications,”
in *Proc*. of ACM *SIGCOMM ‘OI*, 2001.

[18] A. Rowstron , P. Druschel , “Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems,” in *IFIP/ACM International Conference on Distributed System Platform* (Middle-ware), 2001.

[19] http://www.kazaa.com