

叢集定位系統設計與實作

馬詠程¹ 董俊辰² 林彥州³ 林曉楓⁴

¹長庚大學資訊工程系 ycma@mail.cgu.edu.tw

²長庚大學資訊工程系 poiop52@gmail.com

³長庚大學資訊工程系 carpathia705@hotmail.com

⁴長庚大學資訊工程系 allen_790811@hotmail.com

摘要

本論文研究叢集定位系統的設計，以提供智慧型診療服務。此系統透過無線設施定位病患的位置，以提供門診時間提醒以及流感的監控。為了應付大量的病患，我們以叢集系統 (clustered system) 來追蹤病患的位置。在叢集定位系統中，每位病患將被分派給一台 location server 來進行位置追蹤。透過雜湊函數來分配病患給 location servers 以達到負載平衡。在每台 location server 中，病患的軌跡被存放在一個可高效存取的已 hash 的 4-level linked list。我們以多執行緒 (multi-threading) 的方式進行實作，以有效率的處理來自網路的位置封包以及查詢要求，並建立雛形系統(prototyping)以展示系統功能。

關鍵詞：叢集系統、雜湊、多執行緒

1. 前言

近年來大型醫院的病患人數非常大量，以長庚醫院為例在全台共有六個院區，平均每天的門診人數約三萬人，對現有的服務系統而言如此大量的看診人口是個很重的負擔，也是主要想改善的問題。

另外，病患掛號後，需要在候診室等候叫號，因每個人看診時間不同所以等待的時間難以預估，所以大部分的病患及陪同民眾都選擇在候診室等待，非常的耗時，而且大量的門診病患都在候診室等待時，也容易增加疾病交叉感染的機會。

因為大型醫院都占地廣大，病患的行蹤難以掌握，候診的時間又長，當民眾想到醫院的店家逛一下，以消磨等待的時間時，醫護人員無法主動對接近看診時間的病患作提醒的動作，使民眾常常錯過叫號時間而必須重新排隊，而花費更多時間，也使病患產生煩躁情緒，增加醫護人員的困擾。

醫院內部並無提供個人定位的服務，而對於有傳染病疑慮或是行動不便以及需要長時間看護的病患無法進行追蹤、警示、通報等措施，而延誤了緊急救治的時間，形成公共安全的漏洞。

為了解決前述的就診問題，我們擬設計一叢集定位系統，以提供病患就診服務。當病患到有定位系統的醫院看診時，我們分配給病患隨身配戴具備無線感應功能的行動定位標籤(tag)。在院區內，病患可以任意行動，不用一直待在候診室。快到號時，系統會

對病患發出提醒訊息。看診結束就將 tag 還給醫院，來完成整個服務。

此定位系統的環境布建方式如下圖 1 環境布建示意圖。在走廊上，我們以 RFID 天線的感應直徑為間隔距離架設天線，以達成無死角定位。此外，我們於各房門入口處架設天線，以掌握使用者所在房間位置。若空間較大時可考慮在房間中再架設天線。所有的 RFID Reader 皆透過區域網路將感應資訊傳到叢集定位系統存放。

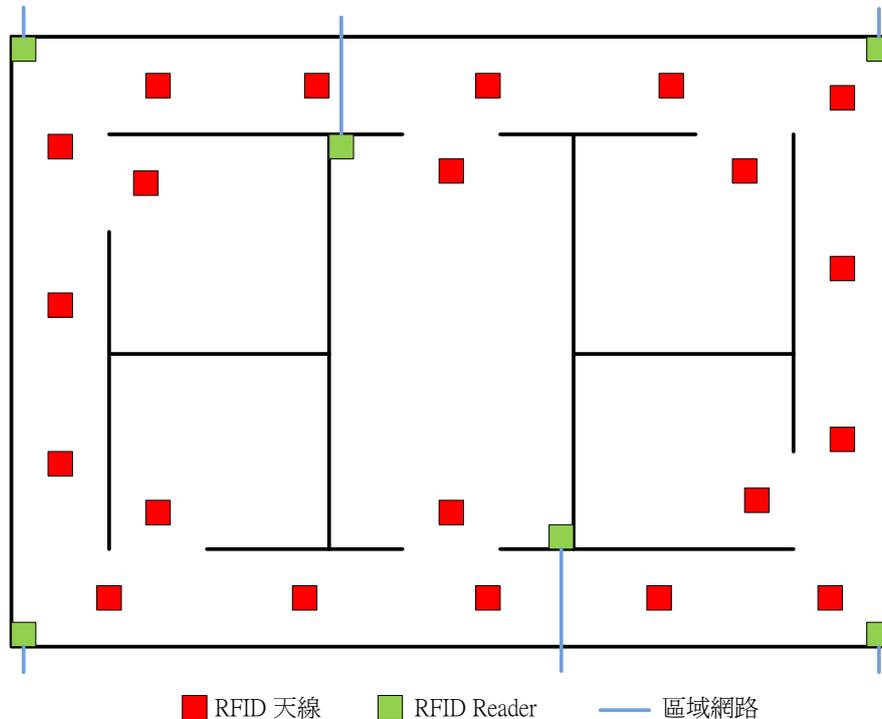


圖 1 環境布建示意圖

1.2 相關研究

- 長庚醫院手機網路掛號與簡訊提醒：

隨著網際網路的迅速發展與無線傳輸的技術提升，長庚醫院也與中華電信合作，可利用手機上網掛號與提醒的功能，病患只要是使用中華電信的門號，連線上 emome，就可執行長庚醫院網路掛號的動作；雖然方便，但此做法的缺點是只能提醒病患就診的日期，但無法告知準確的時間。

- 目前的定位技術，RFID、802.11b/g 的比較：

主動式的 RFID 雖然範圍較廣，但太廣無法準確定位，且 Reader 一台將近十萬元，一個 tag 約一千元成本較高。802.11b/g 範圍更廣，而使用者攜帶的裝置微 Wi-Fi 卡，耗電量較高無法長時間使用。

被動式 RFID 感應範圍約十公尺，此感應範圍較適合我們在醫院內依空間劃分來架設 RFID 天線，tag 價錢約十元較便宜，因此採用被動式 RFID 作為本研究的解決方案。

2. 系統規劃

2.1 系統架構

使用叢集的伺服器架構，為了要達到分散大量使用者使伺服器覆載平衡的目的我們利用多台伺服器的叢集式架構如下圖 2，然後將使用者平均地分配到所有的 Location server 上，並且持續更新使用者最新的位置，因此除了數台 Location servers 之外還使用了一台 Dispatch server 以及 Reader server。

Dispatch server 的主要功能是將報到的使用者分配到數台 location servers 的其中一台上，所以我們對使用者的 ID 做 hash 來決定要分配哪一台 location server 上。如此的好處是我們可以依照系統的負擔或使用人數來調整 location server 的數量。另外，還需要另一台 Reader server 專門解讀 RFID Reader 所感應到的 tag 資訊，然後再做 hash 得知該 ID 所屬的 server，傳送到叢集式的 location servers 儲存起來。

使用 Location servers、Dispatch server、Reader server 三個部份，如圖 2 叢集系統架構，來建構出此叢集定位系統及其完整功能。

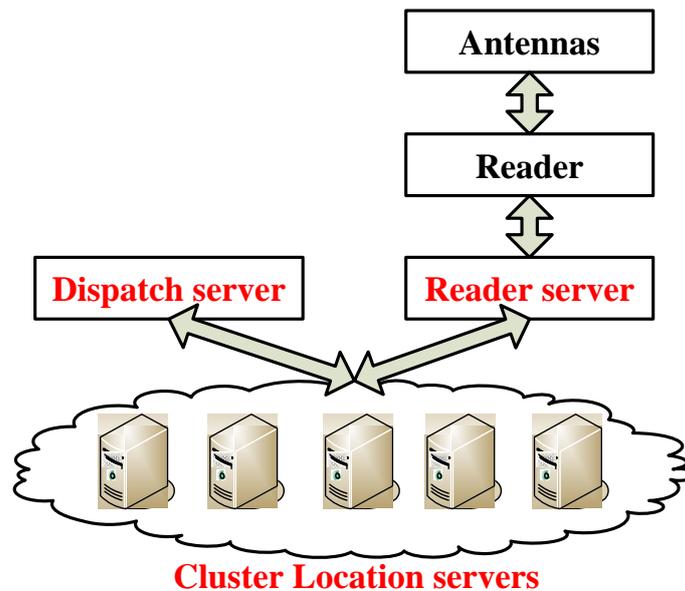


圖 2 叢集系統架構

2.2 系統功能

基於上圖 1 的系統架構以及所要達到的目的，我們可以定出整個系統 5 個主要的功能，如下圖 3 系統功能所示：

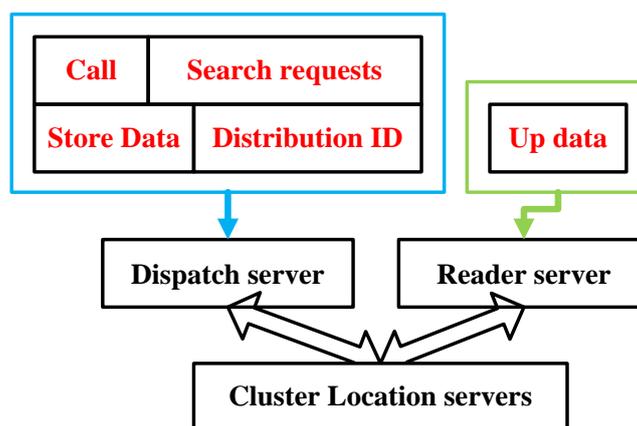


圖 3 系統功能

- (1) Distribution ID：當使用者到醫院報到，醫護人員會發給使用者一個 tag，然後將此使用者的 ID 透過 Dispatch server 新增到 Location servers 上。
- (2) Up data：當使用者報到後，可在院區內任意行動，而系統可透過 RFID Reader 的天線感應到使用者所攜帶的 tag，然後在由 Reader server 透過區域網路傳回 Location server 紀錄。
- (3) Call：當使用者接近看診時間時，會對使用者發送簡訊來提醒，請使用者盡快到候診室。
- (4) Search request：醫護人員可在緊急情況時，透過我們的系統查詢所要找尋的使用者目前的位置。
- (5) Store data：當使用者進入診療室時，醫護人員會回收其手上的 tag，並將使用者的路徑資料儲存到檔案之中後將此使用者及其路徑資料從資料結構中刪除。

2.3 使用流程分析

由系統架構及系統功能，我們可以分析出使用者在院區內的動作流程如下圖 4 use case 所示。首先，使用者現場掛號或是預先掛號然後進行報到，報到時醫護人員會發給使用者一個 RFID tag，然後將使用者 ID 新增到系統之中，然後使用者就可以在院區內任意行動，當使用者在院區內移動時系統會紀錄使用者到過的位置。醫護人員可依需求即時查詢系統中所有使用者目前的位置。當使用者看診時間快到時，系統可以對使用者送出手機訊息通知使用者即刻到候診室等候看診。看診時，透過醫護人員回收 tag，並將使用者的路徑資料儲存起來，在將使用者及其資料重系統中刪除。

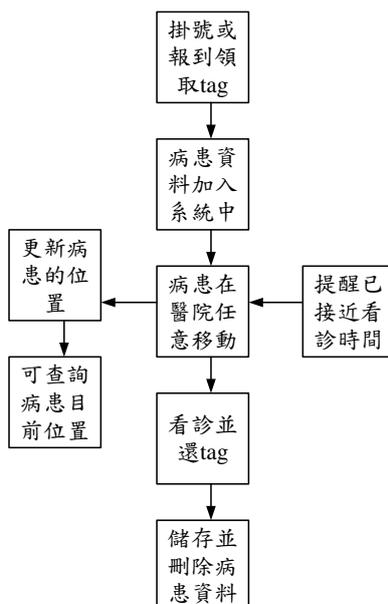


圖 4 use case

2.4 系統通訊機制

系統如圖 1 所示，主要分成三個部份 (1)Location server (2)Dispatch server (3)Reader server，而此三部份彼此是需要做資料的傳輸的，因此我們需要明確的訂定出我們所使用的通訊方式如下圖 5 通訊機制示意圖所示：

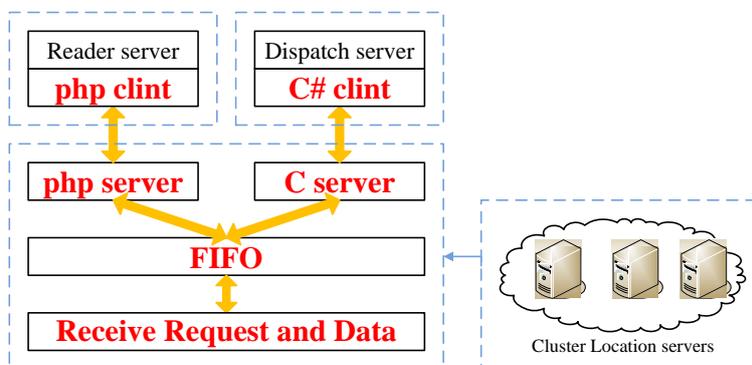


圖 5 通訊機制示意圖

Dispatch server 上主要是運作一個 web page 給醫護人員使用，而我們是使用 PHP 來製作此 web page，因此 Dispatch server 與 Location server 之間就用 PHP Socket 來建立通訊。同樣的 Reader server 是使用 C#來撰寫所以在 Reader server 也是透過 Socket 與 Location server 作資料溝通。

而從 Dispatch 及 Reader server 所傳送到 Location server 的資料是透過 FIFO 的 IPC 機制與主程式作資料的傳遞。我們利用此通訊機制來將所有的程序串聯出完整的系統功能。

3. Location server

第三章介紹建構 Location server 的主要機制及資料結構，我們利用了 multi-thread、multi level hash 來提高效能，使用 link list 來存放所有資料減少磁碟存取次數以提高處理速度，進而提高整體效能。

3.1 Multi-thread

我們設計叢集式系統，主要目的是為了可以容納大量的使用者並降低單一伺服器的負擔進而平衡多伺服器的負擔，而當大量的使用者在院區內移動時會產生大量的位置更新工作，同時還要作其他新增 ID、查詢位置等等 use case，這時系統效能可能無法快速消化這些工作，當接收工作請求很頻繁時之前的工作還沒做完容易造成塞車的情況，因此利用 Multi-thread，如下圖 6。

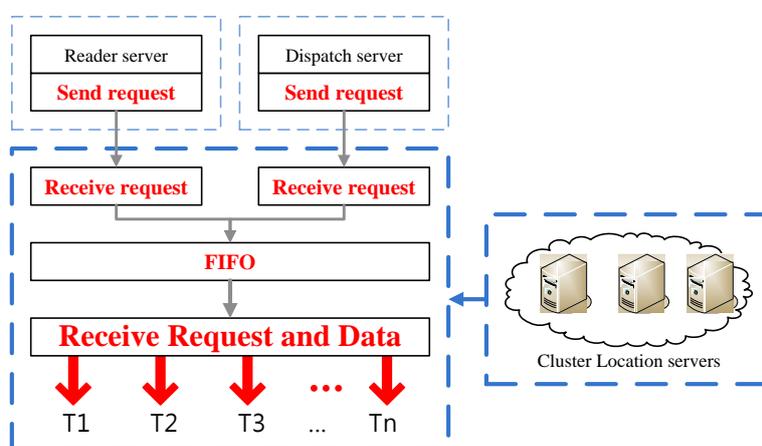


圖 6 Multi-thread

當接收到一個工作請求後就新增一個 thread 來做，主程式就可回到準備接收新的工作請求的狀態。我們利用此方式來處理大量的工作請求，以防影響效能。

3.2 Two level hash

在 Location server 中被分配到同一台 Location server 的使用者 ID 依然很多，對 ID 做 hash 時也很可能會有碰撞的情況發生，所以我們利用了 Two level hash 如下圖 7 Two level hash，對 ID 作兩次 hash，藉此提高分散度來降低碰撞的機率，因為碰撞越少搜尋 ID 的效率就越高。

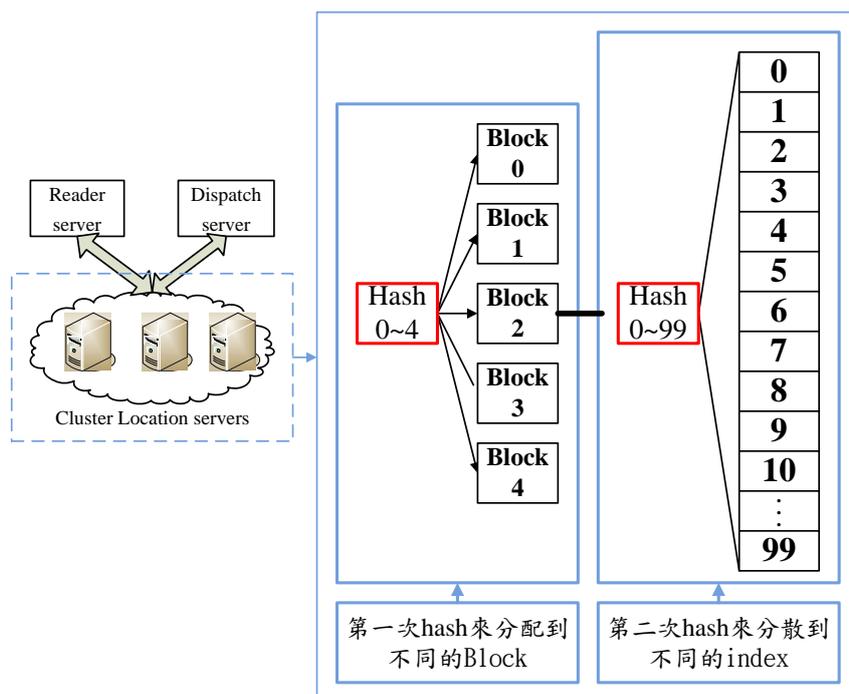


圖 7 Two level hash

3.3 Four Level Link List

系統需要供給大量的使用者存放位置資訊，因此系統存放的資料位置勢必會被頻繁地存取及修改，若是使用一般資料庫的方式來存放，將對硬碟產生大量的存取可能會影響，因此我們希望利用 Link List 為資料結構來存放資料，以減低系統的負擔，進而提高整體的效能。

搭配 two level hash 的想法，我們將 Link List 的結構分為四個部份，如下圖 8 Data Structure 所示：

- (1) First hash list：這個部份是做第一次 hash 時決定要進入那一個 First Level Block 的 index list。
- (2) Second hash list：當決定進入 First Level Block 後再做第二次 hash 時決定要進入那一個 Second Level Block 的 index list。
- (3) ID List：進入 Second Level Block 後往下鏈結的是 ID Node 的 list 結構。
- (4) Data List：每個 ID Node 都會再往下鏈結自己的 Data Node 的 list 結構。

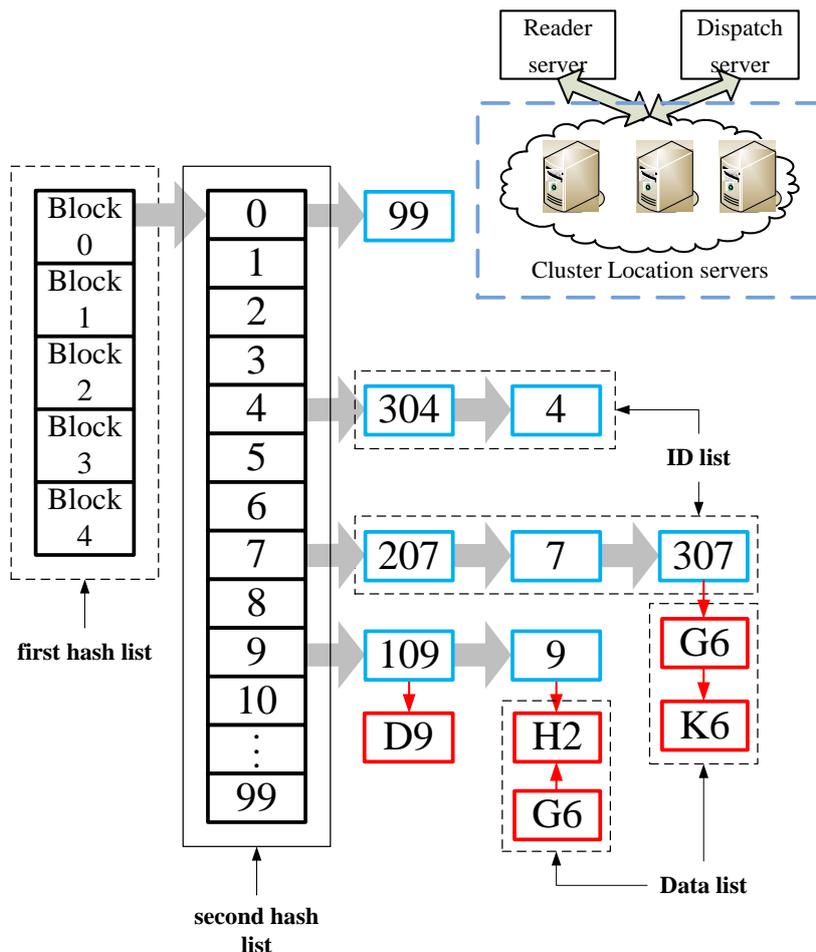


圖 8 Data Structure

4. 模組設定

由第二章的系統架構、第三章的 Location server 設計，我們可再對整個系統做更完整的模組及 method 設定，並以所使用的資料結構來做為切割的基準。我們共分為五個模組：

4.1 Dispatch Module

利用網頁的方式讓醫護人員可進行新增、查詢的工作，且平均分配使用者到 Location servers，如下圖 9 所示。

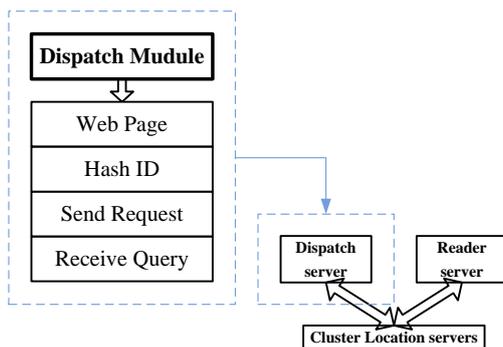


圖 9 Dispatch Module

4.2 Sensing Module

連接 RFID 的 Reader 去接收 Reader 所感應到的 tag 資訊，以及接收所要提醒的使用者的電話號碼，然後對使用者送出提醒的訊息，如下圖 10 所示。

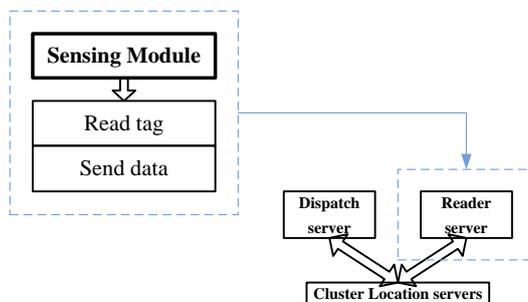


圖 10 Sensing Module

4.3 Receive Module

接收來自 Dispatch server 及 Reader server 所送來的工作請求，如下圖 11 所示。

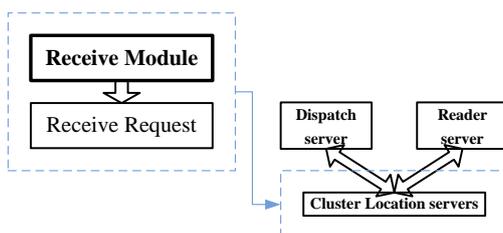


圖 11 Receive Module

4.4 Data Base Module

此模組又分了兩個子模組分別為 Store data 以及 Link List。Store data 子模組是將使用者的路徑資料儲存到檔案之中。Link List 子模組即是本系統的核心，此模組主要是對接收近來的資訊依工作請求來操作 Link List 結構，可進行插入、查詢、刪除，如下圖 12 所示。

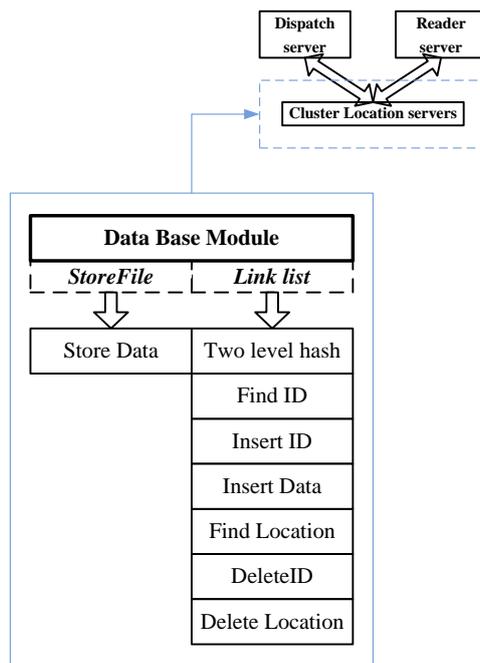


圖 12 Data Base Module

4.5 Query Module

主要是將查詢要求所找到的 location 資訊利用我們所建立的 table 中找出所對應的空間，然後回傳給使用者知道。另一方面將找出之使用者的電話傳給發送提醒訊息的 method，如下圖 13 所示。

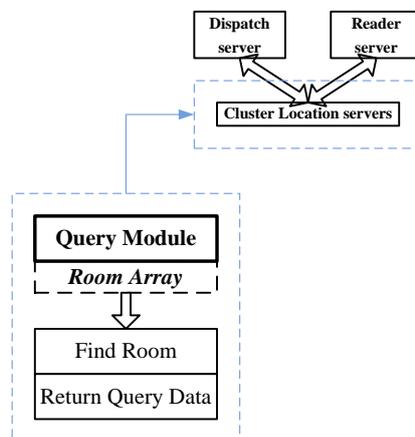


圖 13 Query Module

Dispatch Module 是在 Dispatch server 上運作，主要的目的是要提供醫護人員一個網頁介面來操作，因此也可透過手裝置來查詢位置。Sensing Module 是在 Reader server 上運作，目的是將接收感應的資訊的工作獨立出來，讓 Location servers 的工作更單純化，當需要更換不同的定位設備時 Location servers 依然可以使用。Receive、Data Base、Query 三個模組是架設在 Location servers 上運作來完成整個系統的功能。

5. 系統 Use Case 分析

在第四章所制定好所有系統模組後，需要確認每個模組的 method 可以完成系統所有的功能，因此對系統的五種 use case 作分析，然後再與第四章的模組作比對以確認所有的模組可以完成系統所有的工作。

5.1 新增使用者

醫護人員透過在 Dispatch server 的 web page 輸入要新增的使用者 ID，Dispatch server 會對 ID 作 hash 來決定要分配到哪台 Location server 之後就送出到該 Location server。Location server 接收到後再寫入 FIFO 檔中。此時我的主程式會即時將寫入 FIFO 檔的資料讀取出來，然後對 ID 做兩次 hash 來決定要 Insert 的 first hash list 及 second hash list，最後將 ID 新增到 Link List 之中，如下圖 14 所示。

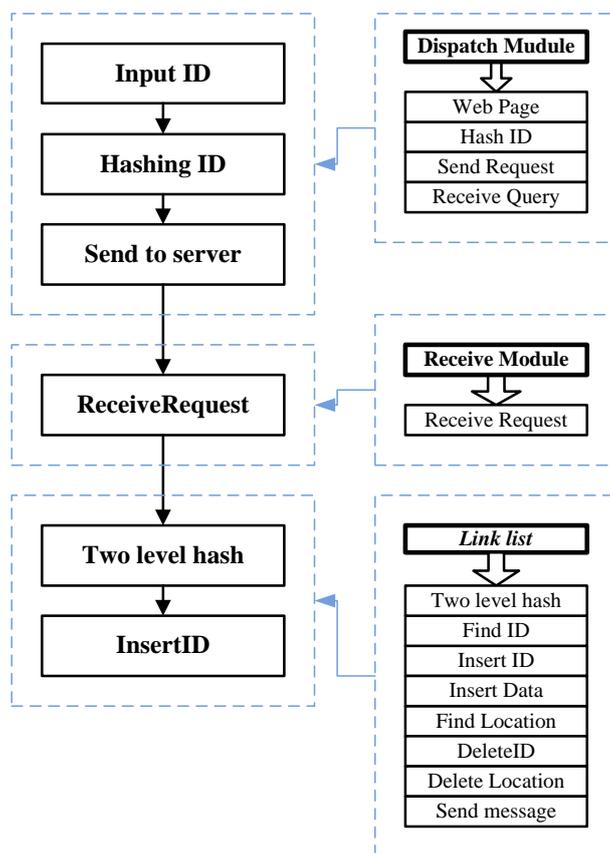


圖 14 新增使用者

5.2 資訊更新

天線感應到使用者身上的 RFID tag，透過 Reader server 上的應用程式讀出 Reader 上的資料後將我們所需要的資訊整理成一字串後，對 ID 做 hash 來得知要送到哪台 Location server，再透過 socket 傳送到 Location server 並寫入 FIFO 檔，主程式會在即時

地將工作請求資訊讀出來。再來，因已經新增 ID 了，所以透過 two level hash 找到所要更新資訊的 ID，然後在將資訊 Insert 到該 ID 所屬的 Data list 的末端。如下圖 15 所示

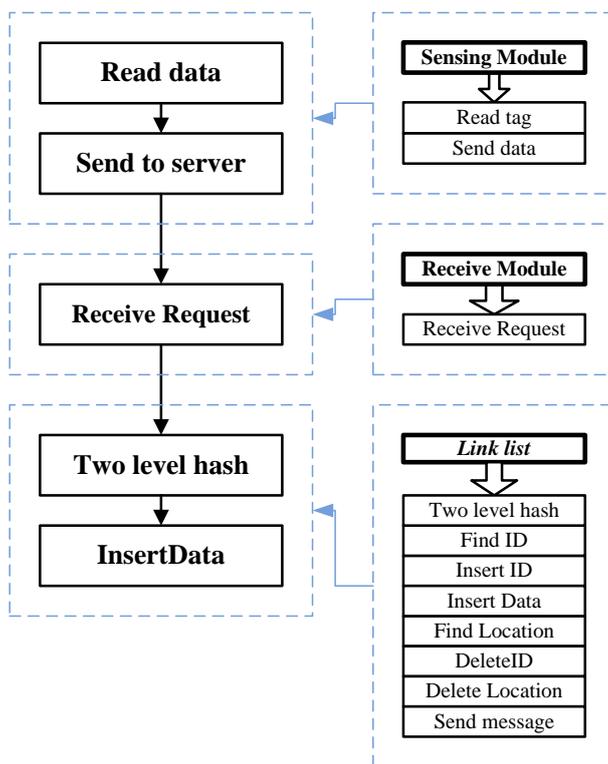


圖 15 更新資訊

5.3 查詢位置

透過 Dispatch server 上的網頁輸入所要查詢的 ID，然後透過 socket 對 Location server 送出查詢的工作請求，系統會從該 ID 下的 Data list 找出最後一筆的位置資訊後在從對應 position 的 table 中找到所在該 ID 的 position，再透過 FIFO 將 position 傳給 socket 後回傳到 Dispatch server 然後顯示給查詢者看。如下圖 16 所示

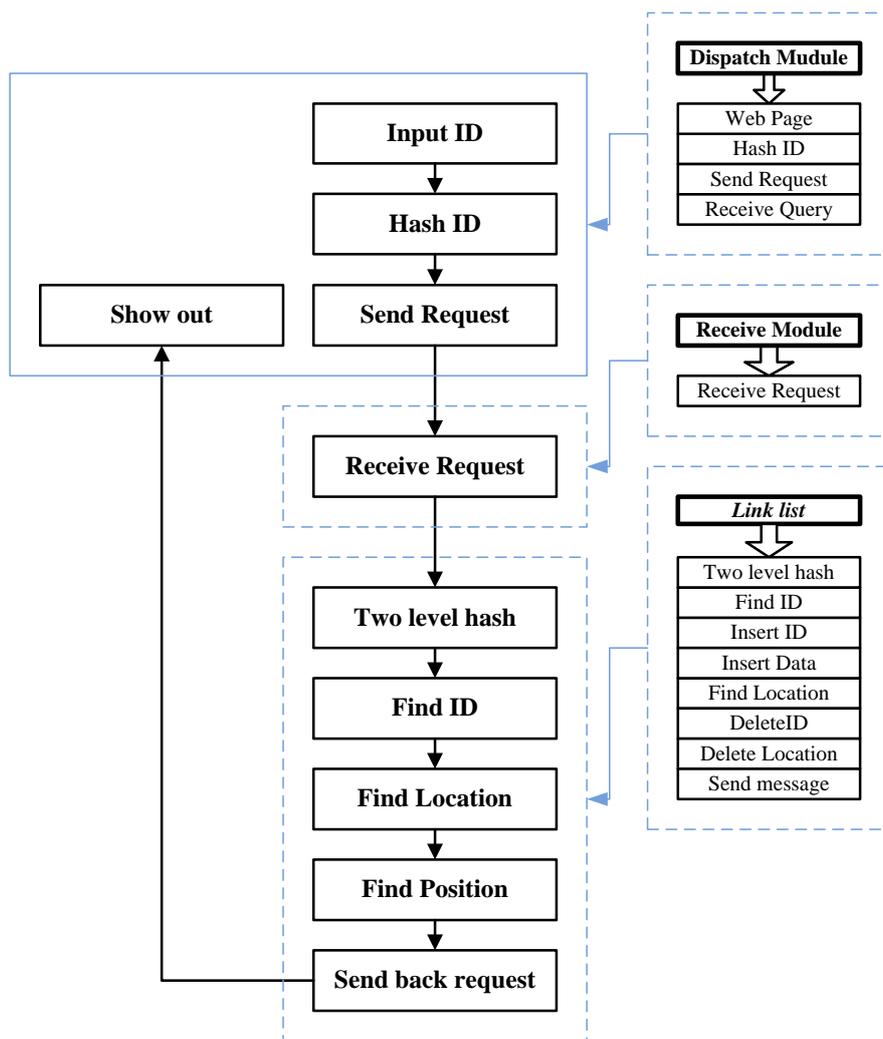


圖 16 查詢位置

5.4 送出提醒訊息

同樣透過 Dispatch server 輸入所要提醒的使用者 ID，然後再將提醒工作請求送到 Location server 上並且從 Link List 結構中找到此 ID，然後將電話號碼取出來，之後將電話號碼傳送到 Reader server 上，再銜接電信業者對使用者發出提醒訊息。如下圖 17 所示

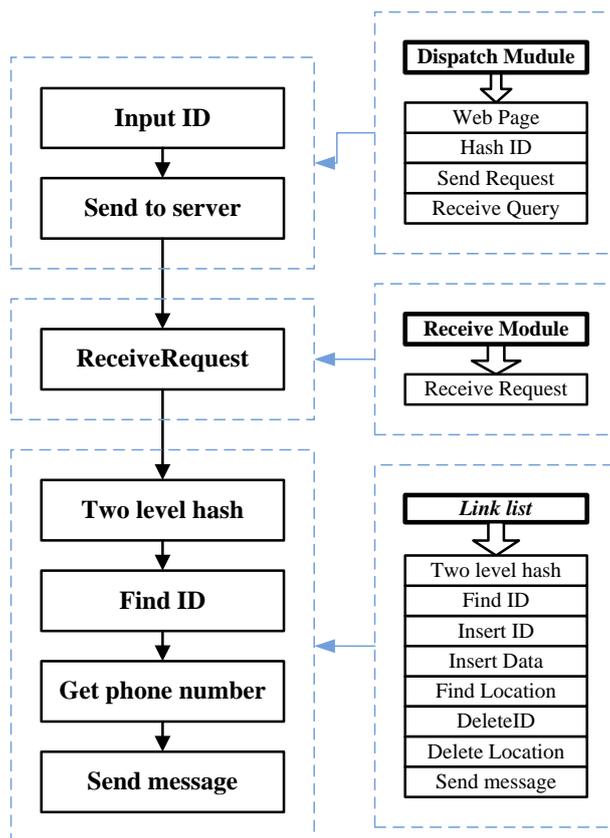


圖 17 送出提醒訊息

5.5 儲存後刪除

當使用者進入診療室時，醫護人員會回收 tag，並透過 Dispatch server 上的 web page 輸入所要儲存並刪除的 ID，然後 Location server 接收到此儲存並刪除的工作請求後會先找到此 ID，再來是將該 ID 所屬的 Data list 上的資料都存到檔案中。最後，將此 ID 及其所有的資訊都從整個 Link List 中刪除。如下圖 18 所示

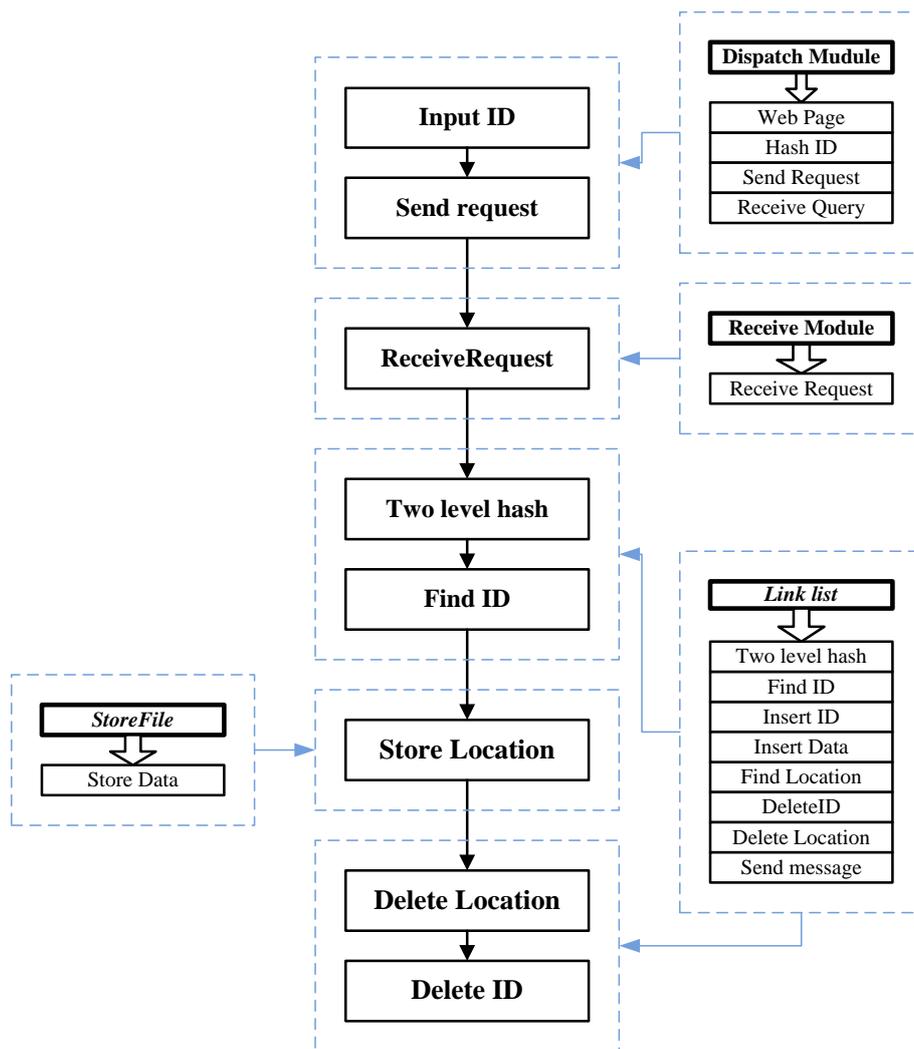


圖 18 儲存後刪除

6. 系統實驗

本章節將對本系統進行執行時間的測試，並對系統效能作評估。

6.1 測試環境

測試的硬體配置如下：

(1) Location Server 一台

- 硬體：Intel Core 2 Quad CPU Q8200 2.33GHz，4G 1066MHz main memory。
- 作業系統：Linux fedora 14 64bit

(2) Send Request Server 一台

- 硬體：Intel Core 2 Quad CPU Q8200 2.33GHz，4G 1066MHz main memory。
- 作業系統：Linux fedora 14 64bit

6.2 測試方式：

由 Send Request Server 產生不同數量的 requests，並透過 socket 連線傳送 requests 到 Location Server 上。在 Location Server 上使用一個 Queue 來接收所有的 requests，當所有存放完所有的 requests 後，再一次讓系統執行。期間只計算執行的時間，而不考慮接收的時間。

6.3 測量結果

表 1 Average Time

Request Number(次)	Execution Time(秒)	Average Time(微秒)
10000	0.387s	0.0387ms
15000	0.591	0.0394ms
16000	0.726	0.045375ms
17000	0.816	0.0480ms
18000	0.822	0.0456ms
19000	1.002	0.05273ms
20000	1.027	0.05135ms

6.4 討論

(1) 由表 1，可以得知系統在此硬體上，執行 19000 個 requests 就會超過一秒鐘，所以可以知道此系統每秒鐘可執行將近 19000 個 requests，因此當 work request 達到 19000 時將需要增加 Location Server 的數量。

(2) 如圖 18 所示，系統會因所要執行的 requests 越多，而平均執行時間越長。

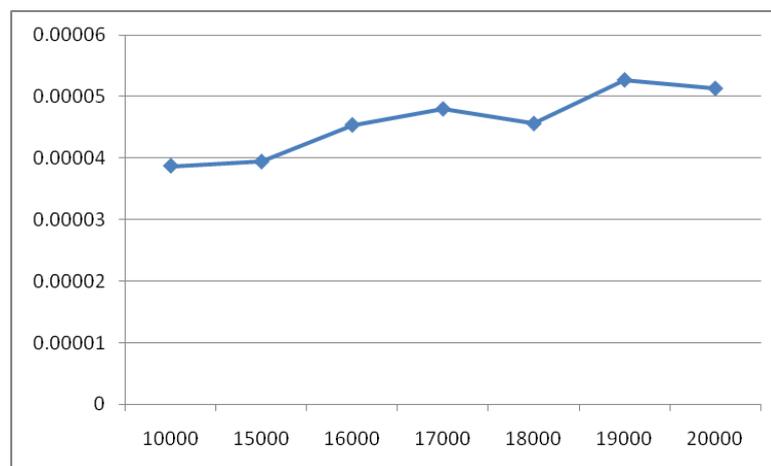


圖 19 Average Time

7. 結論

系統最後呈現是利用到了一台伺服器來連結 RFID Reader，但當初規劃是希望利用主動式的 tag，在發給使用者時就可把使用者被分配到那一台 Location server 寫入 tag，讓傳回的位置資訊封包可以透過 Gateway 直接送到所屬的 Location server 上，並且可直接讓 tag 發出聲響來做提醒。後來改為用 RFID 來做定位，而呼叫的功能則改為用手機簡訊來提醒，相對較不直接，但最終的系統讓 Location server 的功能單純化做儲存、查詢的工作，接收位置資訊的工作由另一台伺服器專門來負責，這樣不僅可以減少 Location server 的負擔還可套用於各種不同的定位設備。

當初設計成叢集式系統的主要目的是為了能容納非常大量的人口使用，並且平衡所有的伺服器的負載，還可增減伺服器的數量，最初規劃時也希望可以設計成 Location servers 之間可以互相溝通的機制，當其中一台伺服器工作量較大時可以將其中一些使用者丟給其他伺服器。但 tag 無法知道回傳的資訊要送到新的伺服器上，而且若當系統在大量紀錄位置資訊時還要將資料在伺服器之間互傳勢必會影響效能。

規劃設計整個系統到實做出來，雖非很複雜，但因為是要容納大量的使用者，所以高穩定性以及效能成為在設計時所要掌握的大原則，因此利用了叢集的想法、multi-thread 以及多階層的 link list 架構來實做。可嘗試利用別資料結構來取代多階層 link list 架構來提高效能。

再進行系統效能測試時，當大量的 requests 要送到 Location Server 時，單一 port 每秒鐘約只能接收 600 個 requests，因此才使用 Queue 先累積 requests，再給系統執行，因此網路可能會是個瓶頸

參考文獻

1. JN-RM-2001-Integrated-Peripherals-API-2v3 (Jennic)
2. JN-AN-1049-Developing-With-High-Power-Modules-1v3 (Jennic)
3. JN-AN-1037-Sleep-Modes-1v2
4. JN-DS-JN5121-1v8 (jennic)
5. iTracerEasyKit_UserManual(Fountains of ubiquitous innovations)
6. Wireless Sensor Networks Research in Taiwan (Yu-Chee Tseng, IEEE VTS Taipei Chapter)
7. What You Should Know about the ZigBee Alliance(Sensors Expo24 September 2003Anaheim Convention Center, Anaheim CA)
8. Michael Keating "Low power Methodology Manual For system-on-Chip Design"
9. A True System-on-Chip solution for 2.4 GHz IEEE 802.15.4 / ZigBee
10. Ron Olexa "Implementing 802.11, 802.16, 802.20Wireless Networks"
11. 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver

Clustered locationing server design and implementation

Yung-Cheng Ma¹ Jun-Chen Dong² Yan-Chau Lin³ Xiao-Fong Lin⁴

¹Department of Computer Science Chang Gung University ycma@mail.cgu.edu.tw

²Department of Computer Science Chang Gung University poiop52@gmail.com

³Department of Computer Science Chang Gung University carpathia705@hotmail.com

⁴Department of Computer Science Chang Gung University allen_790811@hotmail.com

Abstract

This thesis studies the design of clustered location system aimed at providing intelligent clinic service for Chang-Gung Memorial Hospital (CGMH). The clinic service system locates patient positions through wireless facilities to provide clinic time alarm and flu-decease control. To cope with large amount of patients in CGMH, a clustered system is designed to track patient positions. In the clustered location system, each patient is assigned to a location server for position tracing. Load balancing is realized by applying a hash function to assign patients to location servers. In each location server, patient traces are stored in a hashed 4-level linked list for efficient access. Multi-threaded programming is applied to efficiently process positioning packets and querying requests from the network. A prototyping is established to demonstrate the system.

Keywords: clustered system 、 hashing 、 multi-threading