

智慧型裝置於雲端運算之應用-以餐廳搜尋為例

江茂綸
朝陽科技大學
資訊與通訊系

mlchiang@cyut.edu.tw

潘信宏
亞洲大學
資訊多媒體應用學系

wincenpan@asia.edu.tw

張立傑
朝陽科技大學
資訊與通訊系

s10130604@cyut.edu.tw

吳致璋
朝陽科技大學
資訊與通訊系

gguwahu@yahoo.com.tw

蔡明田
朝陽科技大學
資訊與通訊系

s9830018@cyut.edu.tw

詹榮哲
朝陽科技大學
資訊與通訊系

s9730014@cyut.edu.tw

楊士毅
朝陽科技大學
資訊與通訊系

roy4203@yahoo.com.tw

摘要

近年來，智慧型手機已成為風行和有吸引力的議題。每個使用者可以隨時透過智慧型手機來獲得線上服務。然而，眾多的網路服務需求，會使得提供服務的伺服器負擔過重。因此，透過一種包括大量的服務節點與儲存能力及高速網路型態的新型分散式計算環境，雲端運算，是被提出來提供行動用戶大量的網路服務。

因此，本研究將使用雲端運算平台，Hadoop，整合 Android 智慧型手機來提供軟體即服務，如餐廳搜尋服務。此外，即使有大量的搜尋資料存在，每個使用者依然可簡單地透過我們所提供的雲端系統平台來獲得高速的運算速度、大量的儲存空間、高容錯及高擴充性等特色。

關鍵字：雲端運算、分散式系統、Hadoop、Map/Reduce、Android

Abstract

In recent years, the smart phone is a popular and attractive issue. Each user can use the mobile devices to obtain Internet service in any time. However, a huge amount of service demands can make web service overload. Therefore, a new kind of distributed computing environment, cloud computing, which consists of a huge number of service nodes and memories, high-speed networks, and various application services, is proposed to provide a lot of services over the Internet for mobile users.

Therefore, in this study, a cloud computing platform, Hadoop, is used to

integrate android smart phone to provide software as a service, restaurant search service. Besides, each user is easy to obtain high-speed computing, huge storage, high fault tolerance, and scalability by our cloud system platform even if a large number of search data exists.

Keywords: Cloud computing、distributed system、Map/Reduce、Hadoop、Android

1. 前言

隨著 E 化的進展，應用服務的需求也不斷的增加，而硬體設備擴充的速度遠不及服務需求增加的速度。此外，不斷擴增的伺服器將需要更多的空間、電力、網路頻寬及人力等成本。這些衍生出來的成本，將造成一筆龐大的負擔，因此，如何有效的整合目前已有的資源來達成綠色節能環境，將是目前很重要的議題。而雲端運算[1][4][9]將可以透過伺服器的虛擬化及整合，來有效的提供較大的運算服務能力。因此，本文將使用目前最風行的雲端運算建構軟體 Hadoop[1][7]，來針對校園環境來提供大量儲存空間、高容錯、高擴充性及節省能源耗費的雲端運算服務提供平台服務。

此外，近年來因智慧型手機[5]的盛行，且行動上網的普及，愈來愈多的使用者都可以隨時隨地的享受網路的方便性。然而，眾多的網路服務要求，也使得提供服務的伺服器負擔過重。為了同時滿足更多服務要求及降低成本，雲端運算就成目前最主要的解決方案。

因此，本文將結合雲端運算平台 Hadoop 及 Android 手機，以餐廳搜尋服務為例，建置雲端運算軟體即服務平台，提供使用者不但可以在大量的資料中進行快速搜尋及定位其所需要的服務外，更可透過使用者的條件來搜尋眾多餐廳的詳細資料跟菜單。因此，比對的資

料量將遠超過以往提供搜尋服務所要比對的資料量，並對系統造成大量負擔。然而，本文將使用搭配雲端運算服務架構的分散式運算資料庫 Hbase[2]，並透過 Map/Reduce[8][11]的程式分流，將可順利解決海量資料的問題，進而提供較高的運算能力及容錯能力。

本文第二節為文獻探討，首先說明目前所使用的雲端技術之探討；而在第三節中，將說明本研究所提出之系統架構及實作案例；最後，第五節為結論。

2. 文獻探討

2.1 Hadoop 軟體

Hadoop[1][2][5][6][7][8][10][11][12] 是一套讓使用者可以撰寫並執行海量資料應用程式的雲端運算軟體平台。不但擁有儲存與處理大量資料的配置能力外，還可透過分散式檔案系統的幫助，讓一般 PC 可以架設叢集環境，進取得較大的運算能力。此外，每個節點每隔一段時間就會互相交換訊息，所以當某節點發生錯誤，也能及時自動取得備份資料。

2.2 HBase 資料庫

HBase[1][2][3]是由 BigTable[10]所衍生出來支援 Hadoop 的資料庫。以圖一來說明，其有別於一般資料庫系統所使用的 row-key 儲存方式，主要是以 column-family 的儲存方式來表現。

一個 column family 就是一個 column qualifier(Column1、Column2)的集合，裡面可有很多組 qualifier，這些 qualifier 可以視需要隨時新增，而不用重新設定整個資料表。

而 HBase 底層則是使用了分散式的檔案系統 HDFS (Hadoop Distributed File System)[5]來進行儲存；主要是透過將一個資料表拆成很多份後，再由不同的伺服器負責該部份的存取，藉此達到高效能的存取效率。此外，HBase 也同時提供 Map/Reduce[11]程式設計模式、Java 函式庫、php 及 Thrift 等介面。

2.3 HDFS (Hadoop Distributed File System)

而本文所使用的分散式檔案系統則是 HDFS，其主要的作法是將分散的儲存資源整合成一個具容錯能力、高效率且超大容量的儲存環境。因此，在 Hadoop 系統中大量的資料

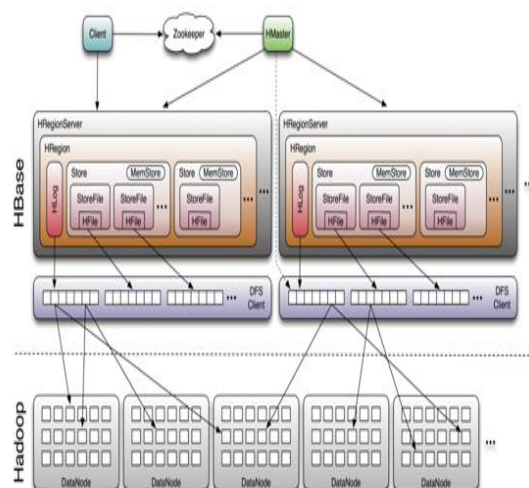
和運算時所產生的暫存檔案，都是存放在這個分散式的檔案系統上。

HDFS 是 master/slave 架構，由兩種角色組成，Name node 及 data nodes，Name node 負責檔案系統中各個檔案屬性權限等資訊，如 metadata 及，namespac 等的管理及儲存；而 data node 通常由數以百計的節點來擔任。通過一筆資料檔會被切割成數個較小的區塊來分別儲存在不同的 data node 上；而每一個區塊也都還有數份複本存放在不同節點中，即使其中一個節點損壞時，檔案系統中的資料還能保存無缺，因此，name node 還需要紀錄每一份檔案存放的位置，以因應存取檔案的需求。而若有節點損壞時，name node 也會自動進行資料的搬遷和複製，其架構如圖二所示。

2.4 Netbean 平台

由於 NetBeans[12]是一個開放框架及可擴展的開發平台，可以用於 Java，C 語言/C++，PHP，Python，Ruby 等程式的開發，並可以通過擴展外掛程式來擴展其功能。

且 Hadoop 主要以 Java 為基底程式語言所開發而成，所以本文就以 Java 為基底的 NetBeans 做為開發平台，以建立一個俱管理者介面的雲端運算服務平台。



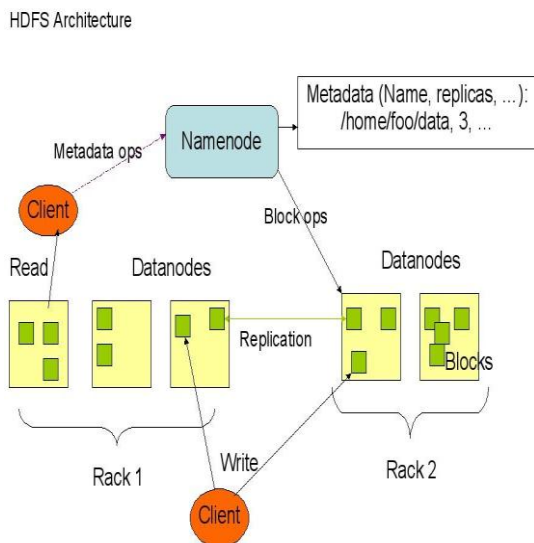
圖一、Hbase 與 Hadoop 資料儲存架構圖

2.5 Thrift

為了解決 Facebook 系統中各系統間大量的傳輸通訊以及系統之間語言環境不同而需要跨平臺的特性。Facebook 提交 Apache 基金會將 Thrift[5]作為一個開源專案。所以，thrift 可以支援多種程式語言，例如：C++、C#、Cocoa、Erlang、Haskell、Java、Ocami、Perl、

PHP、Python、Ruby 及 Smalltalk 等。因此，Thrift 可以在多種不同的語言之間通訊並作為二進位的高性能通訊中介軟體，支援資料(物件)序列化和多種類型的 RPC 服務。

由於 Thrift 適用靜態的資料交換，因此，需要先確定好交換的資料結構。若資料結構發生變化時，則必須重新編輯 IDL 檔，再編譯載入的流程。然而，相對於其它大型系統中的內部資料傳輸 JSON 和 xml，Thrift 無論在執行效能及傳輸大小上都有明顯的優勢。

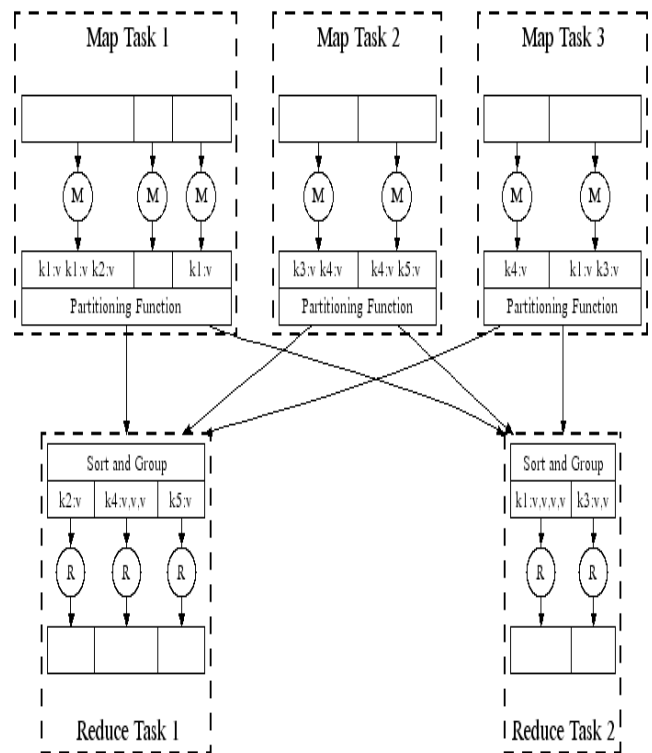


圖二、HDFS 架構圖

2.6 Map/Reduce

而本文所使用的 Map/Reduce[11][1] 是 Hadoop 雲端運算裏的一項關鍵技術。此技術最早是由 Google 所提出來運用在雲端技術中，其主要作法是將要執行的大量資料處理任務，拆解成 Map 和 Reduce 的方式來執行，以達到分散運算的效果。有了 Map 程式之後，就可以使用大量機器分別同步處理及分析每一段 Map 程式，再將每一個 Map 程式所分析出來的結果，透過 Reduce 程式進行合併及彙整出完整的結果。

Map/Reduce 整個架構是由 Map 以及 Reduce 兩個函數所組成，當程式輸入一大群組 Key/Value 鍵值對時，Map 函數會自動將原本的 Key/Value 分拆成許多組中介的鍵/值對，然後，Reduce 函數再合併具有相同 Key 的中間值配對，化簡產生最後的輸出結果，其運作流程如圖三所示。



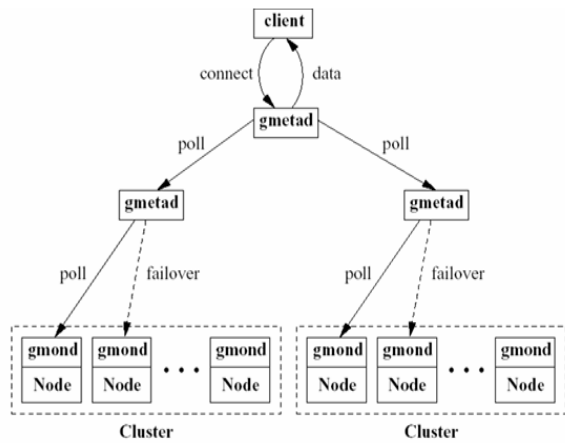
圖三、Map/Reduce 流程架構圖

2.7 Ganglia

本文也使用 Ganglia[2] 監控軟體來監控系統的性能，如：cpu、memory、硬碟利用率、I/O 負載及網路流量情況等。而透過曲線很容易見到每個節點的工作狀態，方便管理者調整及分配系統資源。而 ganglia 是分散式的監控系統，有兩個 Daemon 及 Web 控制端所組成，分別是：用戶端 Ganglia Monitoring Daemon (gmond)、服務端 Ganglia Meta Daemon (gmetad) 及 Ganglia PHP Web Frontend。而本文透過把監控資訊提供給管理者作為管理之依據，這樣管理者透過此軟體與網頁就能輕易快速的知道目前本雲端平台的系統詳細狀況，其架構如圖四所示。

3. 系統架構

本節將說明我們所架構的雲端運算美食搜尋平台。其主要可分為三層架構；網頁使用端、雲端服務端及 Android 使用端，其架構如圖五所示。



圖四、Ganglia架構圖



圖五、雲端運算美食搜尋平台系統架構圖

3.1 網頁使用者端

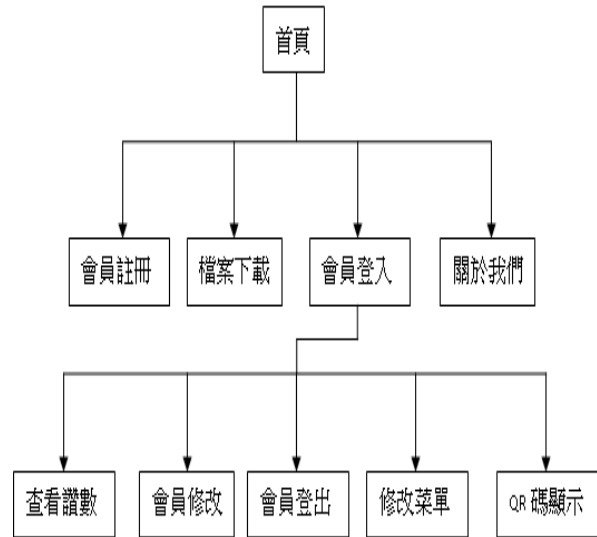
網頁使用者端功能說明：

- 會員註冊：給店家填寫基本資料，並對查詢這家店所在的真實經緯度。
- 檔案下載：我們提供給訪客下載我們手機與平版的 APK 檔案。
- 會員登入：提供經過我們管理者認證，註冊成功的店家可以登入的管道。
- 會員登出：讓已登入的使用者可以登出系統。
- 關於我們：介紹這個管理者的成員與開發動機與感想。
- 查看讚數：店家可以看到，有多少手機使用者針對他的店給過幾個讚。
- 會員修改：給店家修改基本資料。
- 修改菜單：提供店家新增與修改菜單資

訊。

- QR 碼顯示：建立店家專屬 QR 碼，讓店家可自行列印到 DM 等，方便使用者端可以使用 QR 碼來查詢店家的詳細資料。

整體架構如圖六所示。



圖六、網頁使用者端架構圖

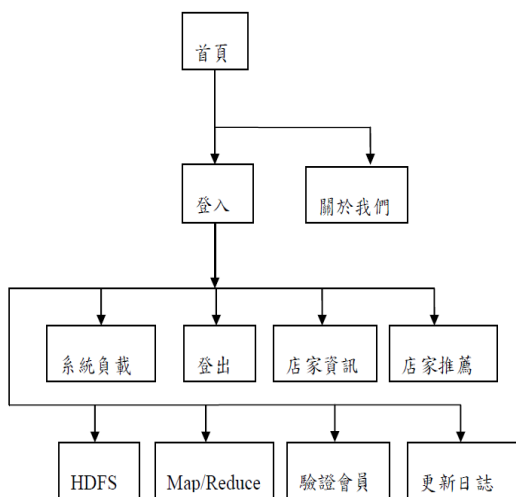
3.2 網頁管理者端

網頁管理者端功能說明：

透過本介面，管理者可以進行網頁驗證會員、推薦店家、查看店家資訊、查看雲端環境運行的程序及節點狀態，來進行雲端環境的設置及管理，其步驟如圖七所示。

- 登入/登出：管理者帳號登入/登出。
- 關於我們：介紹管理成員、開發動機與感想。
- 系統負載：讓管理者看到目前雲端上各節點目前的 CPU、記憶體、硬碟利用率，I/O 負載及網路流量情況等狀態。
- 店家資訊：顯示目前已註冊所有店家的所有詳細資料。
- 店家推薦：透過設置推薦機制，來方便使用者進行選擇。
- 驗證會員：驗證有申請註冊的店家資料。
- 更新日誌：可以看到訪客的登錄、申請註冊及店家的資料修改動作等，將方便管理者隨時管理本雲端運算平台。
- HDFS：管理者可透過本功能看到雲端平台上的分散式檔案系統的剩餘容量跟使用容量等。
- Map/Reduce：給管理者看到所有進行分散式運算的工作之目前詳細狀態。

其整體架構如圖七所示。



圖七、網頁管理者端架構圖

3.3 伺服器端

本雲端運算平台的伺服器架構[1]如圖八所示。

- 雲端應用程式：這裡透過 JAVA 來撰寫服務及與底層 HBase 的資料庫溝通的中介程式。此外，也需根據要求撰寫自動執行分散式運算的工作，並回收執行結果。
- Map/Reduce：本功能可以於伺服器處理交叉式搜尋，把儲存在資料庫裡的資料直接讓每一台節點進行分散式運算處理。
- HBase：利用分散式 NOSQL 資料庫在伺服器中儲存所有店家的資料。
- HDFS：Hadoop Distributed File System (HDFS) 將分散的儲存資源整合成一個俱容錯能力、高效率且超大容量的儲存運算環境。而 Hadoop 系統中大量的資料和運算時產生的暫存檔案，都是存放在這個分散式的檔案系統上，且 Map/Reduce 與 HBase 也是架構在此基礎上。
- 實體機器：在伺服器上使用 VMware 來建置 3 台虛擬電腦，並使用 Ubuntu10.04 的作業系統來建置 Hadoop 叢集雲端環境。

其整體架構如圖八所示。

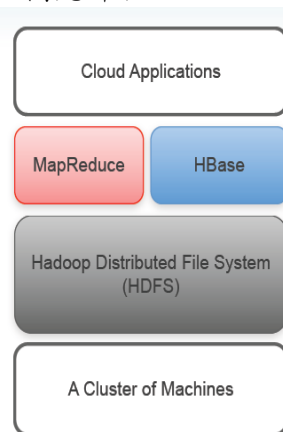
伺服器程式功能說明：

店家透過伺服器架設的網站來寫入資訊，並將店家地址轉換成所在的經緯度後進行定位。往後店家可以透過本平台隨時更新及擴充店內資訊，進而加速資料的傳達及流通。在正式運行服務時，Server 端必須隨時開啟監聽程式，以等待使用者提出服務請求。

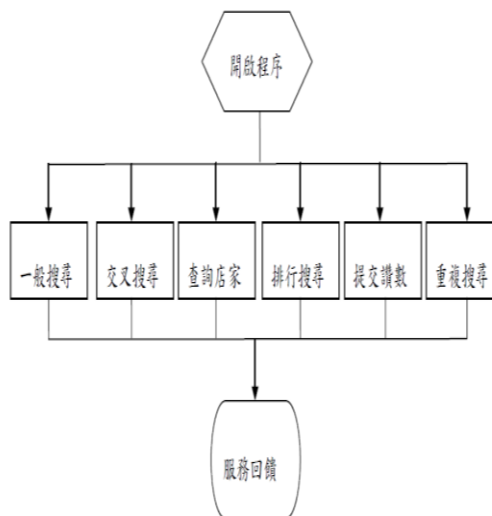
其相關功能描述如下。

- 重複搜尋：在收到關鍵字搜尋時，就去查詢是否有搜尋過這樣的條件，如果搜尋過，就把當時搜尋後的條件送出，以節省再去運算的搜尋時間。
- 一般搜尋：直接對該關鍵字進行搜尋本機搜尋。
- 交叉搜尋：若收到較多關鍵字時，將資料透過 Map/Reduce 程式送至 Hadoop 控制節點，讓各服務節點去處理資料庫裡的資料作分散式運算，再回傳結果。
- 提交讚數：把使用者提出的店家做增加讚數或是減少的動作，在回傳此店目前的讚數。
- 排行搜尋：排出目前擁有最高讚數的前 5 個店家。
- 查詢店家：回傳使用者提出的店家所有資料。
- 服務回饋：透過 JSON 統一交換格式，再將資料送出給提出請求的對方。

其主要架構如圖九所示。



圖八、伺服器整體架構圖



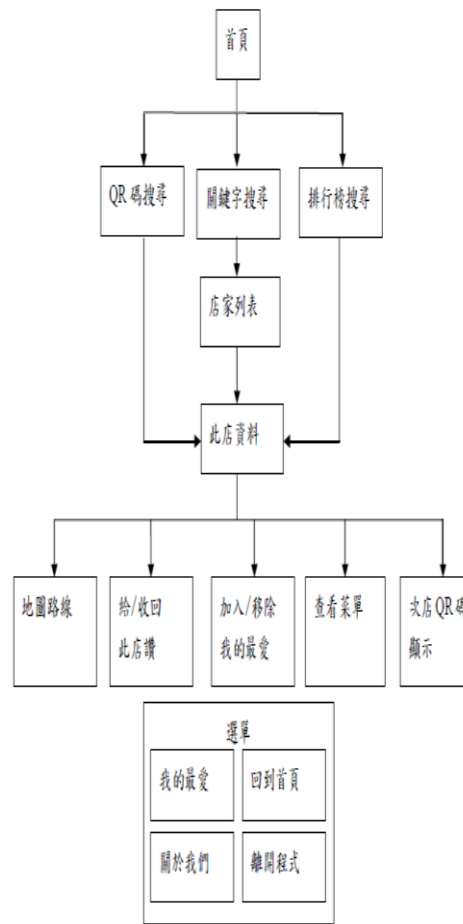
圖九、伺服器程式架構圖

3.4 手機使用者端

使用端 Android 手機程式功能說明：

手機 APP 所提供的美食搜尋服務，將可以透過瀏覽菜單的模式，快速搜尋及定位到所需要的美食地址。此外，我們也有提供多元化的搜尋服務，如關鍵字搜尋、重複搜尋、排行榜搜尋及 QR 碼搜尋等，讓使用者可以更精準的針對餐廳菜單進行大量的篩選與比對，讓使用者能夠找到最符合條件的目標美食資訊。

- 關鍵字搜尋功能：提供四種搜尋條件給使用者做搜尋店家的選擇，並負責接收伺服器所回傳的結果。
 - 排行榜搜尋：回傳目前資料庫裡讚數最高的前五個店家資料給手機使用者。
 - QR 碼搜尋：開啟掃描模式，把掃描的 QR 碼對應資料庫中的店家名稱，再顯示於手機上。
 - 店家列表：顯示符合條件而搜尋到的店家名稱，並以列表方式呈現。
 - 店家資料功能：顯示被點選的店家之詳細資料，例如：電話、地址、營業時間及備註等。
 - 地圖路線：透過定位顯示目前位置與店家的路線規劃。
 - 給/收回此店讚：透過本功能可以給予或收回店家讚，也可看到店家所累計的讚數。
 - 加入/移除我的最愛：把特定店家加入/移除使用者的最愛清單。
 - 查看菜單：可以看到這家店所販賣菜單的詳細資料。
 - 回到首頁：直接回到首頁。
 - 關於我們：介紹這個管理者的成員與開發動機與感想。
 - 觀看我的最愛：看到目前使用者所加入的最愛店家列表，並可點選更新店家的最新資料。
 - 離開程式：完整的直接離開本程式，下次進入就直接是從首頁進入。
 - 店家 QR 碼顯示：顯示店家 QR，並可讓其他手機用戶能透過本 QR 碼取得資料。
- 其整體架構如圖十所示。



圖十、Android 手機程式架構圖

4. 結論

本文所提供之雲端運算服務平台的主要目的是希望提供軟體即服務的功能給店家及使用者來使用，讓餐廳店家可以透過網頁方式將自己的資訊公佈給一般大眾，同時還可以透過本平台及手機 APP 功能來隨時維護及推廣相關促銷資訊。

此外，手機使用者也可透過本雲端運算服務平台所建置的 HBase 快速及便利地從海量的資料庫中找尋符合心目中條件的餐廳(菜單)資訊。此外，本平台提供不同特色的搜尋服務，利用 QR 碼讓店家與使用者方便共享餐廳資訊，並可參考排行榜來進行選擇。

對於管理者而言，配合 Hadoop 與 Ganglia 不但可以快速掌握雲端平台資訊，如處理量或硬體不足等，更可透過與分散式資料庫 HBase 的結合來解決搜尋海量資料的問題，進而提升雲端運算平台的處理效能。

5. 參考文獻

- [1] Jazz, “雲端運算相關技術與應用之初探”, 台北市立教育大學, Mar. 17, 2011. <http://trac.nchc.org.tw/cloud/wiki/TMUE11031>。
- [2] Lengzijian, “Ganglia 的結構和安裝”, 2012, <http://blog.csdn.net/lengzijian/article/details/7102320>.
- [3] Rafan, “HBase 介紹”, Nov. 16, 2008, <http://www.hadoop.tw/2008/11/hbase.html>。
- [4] Wikipedia, “雲端運算”, Apr. 3, 2009, <http://zh.wikipedia.org/wiki/%E9%9B%B2%E7%AB%AF%E9%81%8B%E7%AE%97>。
- [5] 林旻君, “Facebook 到底用了哪些技術?” , Sep. 11, 2011, http://www.syscom.com.tw/ePaper_Content_EPArticleDetail.aspx?id=140&EPID=168&j=5&HeaderName=%E7%A0%94%E7%99%BC%E6%96%B0%E8%A6%96%E7%95%8C。
- [6] 黃植懋, “伺服器虛擬化技術簡介”, 台灣大學計算機及資訊網路中心作業管理, Mar. 20, 2008, http://www.cc.ntu.edu.tw/chinese/epaper/0004/20080320_4012.htm。
- [7] S. Ghemawat, et al., “The Google file system,” Proceedings of the nineteenth ACM symposium on Operating systems principles, New York, USA, October, 2003, pp. 29-43
- [8] T. Gunarathne, et al., “MapReduce in the Clouds for Science,” 2010 IEEE Second International Conference on Cloud Computing Technology and Science (CloudCom), 2010, pp.565-572.
- [9] Weiss A, "Computing in the Clouds," *NetWorker on Cloud computing: PC functions move onto the web*, Dec 2007, pp.16-25.
- [10] Wikipedia, “Bigtable,” Aug. 18, 2011, <http://zh.wikipedia.org/wiki/BigTable>.
- [11] Wikipedia, “Map/Reduce,” Oct. 11, 2011, <http://zh.wikipedia.org/wiki/MapReduce>.
- [12] Wikipedia, “Netbeans,” May 13, 2011, <http://zh.wikipedia.org/wiki/NetBeans>.