國立臺灣大學醫學院暨工學院醫學工程學研究所

碩士論文

Graduate Institute of Biomedical Engineering
College of Medicine and College of Engineering
National Taiwan University

Master Thesis

巨量資料視覺化模型建構之探討
大腸癌盛行率與飲用水質關係

Big Data Visualization System Design and Research of Interaction between Colorectal Cancer and Drinking Water

王浩

Hao Wang

指導教授： 翁昭旼博士
蔣以仁博士

Advisor: Jau-Min Wong, Ph.D.
I-Jen Chiang, Ph.D.

中華民國 105 年 8 月

August 2016

doi:10.6342/NTU201601304

國立臺灣大學碩士學位論文
口試委員會審定書

巨量資料視覺化模型建構之探討

大腸癌盛行率與飲用水質關係

Big Data Visualization System Design and

Research of Interaction between

Colorectal Cancer and Drinking Water

本論文係　王浩 君（r03548047）在國立臺灣大學醫學工程學研究所完成之碩士學位論文，於民國 105 年 7 月 4 日承下列考試委員審查通過及口試及格，特此證明

口試委員：翁昭旼

（指導教授）

陳中明

蔣以仁

張淑惠

所長：陳中明

doi:10.6342/NTU201601304

誌謝

整個系統以及論文完成的當下，學生內心非常激動也百感交集，感謝同學的啟發讓學生能設計這套系統，疾病地圖一直是學生很想做的主題，其中包含分散式系統、視覺化等議題，能夠完成並且受到大家肯定也是與有榮焉，感謝指導教授翁昭旼老師以及共同指導教授蔣以仁老師，對於許多臨床知識的傳授以及系統設計的建議，讓學生能夠順利完成此份論文，整個系統經歷三次改版，每次改版都是對於系統架構以及設計理念的更新，感謝學長對程式設計的訓練，讓學生對原本不熟的 MVC 架構能有清楚的認知，感謝實驗室的夥伴，兩年來的相處大家都很融洽，有問題時總是可以一起討論並解決。

最後，感謝家人的支持與陪伴，沒有家人無怨無悔的付出與陪伴無法順利完成學生的求學之路，兩年的時間裡可以發生很多事情，學到的不僅是學問更是人生，想要得到必須先學會放下，有些人想留也留不住，期待的事情也未必會發生，唯有堅持下去才有機會看到結果。

doi:10.6342/NTU201601304

中文摘要

資料視覺化是本論文的主軸，它也是使用者與資料溝通最直接的方法，利用 D3.js 函式庫建立整套疾病地圖系統，透過互動事件與動畫呈現讓使用者感受資料的特性；透過疾病地圖從空間面向觀察臺灣整體的疾病分布與趨勢；透過疾病趨勢圖觀察特定縣市之時間面向資料，展示疾病各年度間的盛行率、通報數以及平均年齡...等不同資訊，同時系統也提供一個便於分析資料的介面清楚地比較不同縣市的差異；利用非同步技術動態載入環境資料庫(自來水水質資料庫、水庫水質資料庫)，在整合趨勢圖中隨著時間演進泡泡的位置移動與大小的縮放除了可以瞭解疾病的趨勢也可以觀察特定環境屬性是否與疾病有一定程度上的相關。

本研究系統採用國家衛生研究院全民健康保險研究資料庫百萬人抽樣歸人檔做為疾病資料基礎；為了處理如此大量資料而選用 NoSQL 資料庫 MongoDB 做為資料儲存的系統，利用 mapreduce 技術提升在分散式資料庫查詢的效能並能執行較複雜的運算，剔除不符合條件的就醫紀錄並將原本歸人的資料依據區碼歸檔，建立 cache 系統避免頻繁的資料庫伺服器存取，將系統資源做最有效的應用。系統開發採用 MVC 架構，讓系統模組化以增加其擴充性，可依據使用者查詢的疾病代碼(ICD-9)載入適當的預測模組或者功能模組。

關鍵字:健保資料庫、NoSQL 資料庫、疾病地圖、mongoDB、巨量資料、資料視覺化

doi:10.6342/NTU201601304

Abstract

The core of this thesis, Data visualization, is a way of user communication with data. Using D3.js tools to build this disease mapping system, which allows user to feel the change of the data by events selection and animation. With the Disease Map function, users are able to observe the distribution of the disease in spatial aspect. With the Disease Trend function, users are able to read the prevalence, count, and average age etc. of any city in time scope. These functions, also provide a interface to compare data between different cities. Loading environment database dynamically, binding with Hybrid Bubble Chart function by observing the position and the radius change of the Bubbles at different time points and let users be able to feel whether is there any relative trend between environment attributes and the disease occurrence.

We use Nation Health Insurance Research Database (NHIR) as the database of this system which contains medical records of a million patients. In order to deal with this enormous amount of patient data, we select MongoDB, which is a distributed document NoSQL database. With mapreduce technique we can run complicated operations. Eliminating those data which doesn't fit the query condition, then restructure the data by geographical distribution. By using Cache system to keep our database away from busy accessing to increase the query efficiency. We also applied MVC framework to make this system more expendable and able to load specified prediction module or function module depend on the ICD-9 code user input.

Key word: Nation Health Insurance Research Database、NoSQL Database、Disease Mapping、Big Data、Data Visualization

doi:10.6342/NTU201601304

目　錄

doi:10.6342/NTU201601304

doi:10.6342/NTU201601304

doi:10.6342/NTU201601304

圖目錄

doi:10.6342/NTU201601304

表目錄

doi:10.6342/NTU201601304

第一章　緒論

1-1 研究背景與動機

在數位化的今天，我們習慣使用電腦來儲存資訊，這些資訊成為可觀的資料庫，而當這些資料正以極快的速度膨脹時，資料庫與資料庫中的數據是否帶有許多關聯呢？

隨著科技日新月異，巨量資料的應用漸漸受到重視，視覺化是使用者與資料溝通最重要的方法，透過視覺化平台將資料庫結構化的資料用適當的方式呈現給使用者以凸顯資料的特性，近年來在生醫領域更是經常將視覺化作為研究題材，用以探討像是共病研究[1]、食品安全、空氣汙染、水汙染.....等因素對於健康的影響，本研究的主要目的希望能建立一套無論一般使用者以及擁有醫學知識背景的進階使用者皆能容易使用的疾病地圖視覺化平台，透過這個系統將巨量的就醫資料透過系統運算再用地圖的方式呈現，讓一般使用者可以透過這套系統得知特定疾病在臺灣的分布狀況，讓進階使用者可以輕易的從這些圖表中獲得想要的資訊，並能從時間與空間兩個不同角度比較與分析健保資料庫中的資料，整合的環境資料庫也可以作為探討環境因子與疾病關聯性的基礎，搭配適當的預測模型可以對特定疾病的趨勢作更深入的探討。

出自於想探討不同資料庫之間關聯性的動機，選擇了政府機關所提供的健保資料庫與水質監測資料庫做為材料，將其串接來架設一個數據視覺化平台，學生認為疾病的發生趨勢會是具有空間性[2]所以使用疾病地圖(disease mapping)的以地理資訊系統方式呈現，方便學者利用此平台做更深入的相關研究。

doi:10.6342/NTU201601304

1-2 研究動機及目的

本研究的目的以大資料視覺化以及系統架構為主軸，主要呈現疾病在臺灣的分布狀況，衛生福利部疾病管制署傳染病統計資料查詢系統[3]做為發想，使用者可以選擇有興趣的數種疾病對健保資料庫進行查詢，並透過互動式疾病地圖幫助使用者了解特定疾病各年齡層在臺灣的分布狀況，此外可以加上不同環境資料庫的數據透過視覺化的工具讓資料更容易理解，從而讓使用者從就醫資料以及其他數據中探討疾病的現況以及關聯性進而取得有用的資訊甚至是預測未來趨勢。

系統方面，本研究嘗試建立一個適合大量數據視覺化的平台除了希望能挖掘有用的訊息，同時也希望資料的處理是有一定效能的，因此選用 NoSQL 資料庫[4]MongoDB[5]來處理巨量的健保資料，由於期望該系統需要達到即時查詢的效果，所以建立叢集系統以及利用分散式運算(MapReduce)搭配快取(Cache)達到最好的查詢效能。

材料方面選用健保抽樣資料庫、水庫水質監測資料庫以及自來水水質抽驗資料；健保資料庫的基礎是由呈保全民健康保險每天所記錄成千上萬的就醫紀錄所累積而成的，由中央健康保險署所提供的 2010 年承保資料檔以「身份證字號加上生日加上性別」歸人，將身分欄位加密後，交由國衛院製作成「全民健康保險研究資料庫」及各加值資料檔案，自抽樣母群體隨機抽樣，取得 100 萬人樣本，以本研究所使用的健保資料庫來說，從 1996 年至 2010 年，15 年的資料總共包含了超過十億筆就醫紀錄以及醫令紀錄；由於認為大腸癌與水質中的因子可能有一定關聯，故選用水庫水質監測資料庫做為環境資料庫；現代人較少直接飲用河川的水故加上自來水水質抽驗資料作為選項。

doi:10.6342/NTU201601304

1-3 研究流程

雖然網路上現有的的視覺化工具相當多，但直接使用這些工具通常會有很限制與冗餘的功能導致系統非常缺乏彈性，甚至使用不適合資料的呈現方式造成事倍功半的效果，為了建立地圖視覺化工具，所以學生解析政府提供之臺灣地理資訊系統檔案格式 SHP(shapefile)[6]，並研究開源函式庫 D3(Data-driven document)開發系統專屬的疾病地圖系統以呈現疾病在臺灣的盛行狀況，為了完整的呈現資料的狀況，開發這套系統時將資料分為時間面向以及空間面向來呈現，利用 SVG 物件建立充滿互動性並能凸顯疾病趨勢的資料視覺化平台。

考量每位患者的就醫資料依據為數不同的就醫紀錄來存放，傳統關聯式資料庫受限於其關聯式結構而無法彈性規劃、查詢時需要到各表格中抓取資料、表格合併的 join 演算法相當的消耗資源以及其一致性策略...等，將對系統造成龐大的負擔也會導致查詢時花費大量的時間對效能造成很大的威脅，這些因素都使得傳統關聯式資料庫不適用於大數據領域，選用 NoSQL 資料庫 MongoDB 作為資料庫後台，可以解決傳統關聯式資料庫在資料量過大所造成的儲存管理問題，分散式運算架構及無綱要分析(schemaless)的設計，意味著資料庫可以藉由在機器中增加硬碟即可輕易地進行水平擴充，雖然解決的儲存的負擔但也需要更注重資料以及運算負載的平衡也成為著手研究的熱門議題。

系統方面，學生研究 mapreduce 技術在分散式的 mongoDB 叢集資料庫進行複雜的資料分析，將健保資料庫的資料去除不符合條件的就醫紀錄並將承保人利用區碼歸檔作為輸出使用；為了達到更好的使用者體驗，建立快取(cache)系統，避免資料庫頻繁地存取資料庫伺服器而拖垮效能。

doi:10.6342/NTU201601304

第二章　研究材料與相關文獻探討

有鑑於系統的建立需要整合疾病資料以及地理資訊系統(Geographical Information System, GIS)，使用全民健保資料庫作為疾病資料基礎，包含 15 年之健保就醫紀錄抽樣檔，可以作為 15 年內疾病發生的抽樣代表，地理資訊系統將數位化的空間資料與屬性資料整合並具有儲存、查詢、分析和展現…等，透過這些功能讓我們能探討與空間相關的問題，疾病地圖[7]是一種將資訊視覺化的方法，將疾病的發生狀況以向量式資訊透過地圖呈現給使用者。

2-1 健保資料庫

本研究所使用的資料來自於國家衛生研究院，每年中健保署將前一年的健保資料選取可供研究使用的檔案匯出，將身分欄位加密後，交由國衛院製作成「全民健康保險研究資料庫」(National Health Insurance Research Database, NHIRD) [8]。由中央健康保險署所提供的 2010 年承保資料檔以「身份證字號加上生日加上性別」歸人，可得 27,378,403 人之資料，作為資料母檔。在資料母檔中，剔除性別不詳者，選取在保者 23,251,700 人之資料為抽樣母群體。再從抽樣母群體隨機抽樣，取得 100 萬人樣本。以抽樣檔來說，抽樣檔之性別、年齡分布、就醫科別分布與平均投保金額，與母體間均無顯著的差異。

此原始資料為全民健保資料庫 1996-2010 年一百萬人之承保抽樣歸人檔(Longitudinal Health Insurance Database 2010, LHID2010)，有 25 組資料，每組有 4 萬人，共 100 萬人歸人檔紀錄，研究資料年度為 1996 年至 2010 年，單一歸人檔之結構為「承保資料檔(ID)」、「生日(id_birthday)」、「居住區碼(id_rec.reg_zip_code)」、以及數筆「就醫紀錄(func)」而根據使用者所輸入的疾病代碼抽取資料庫中具有特定疾病代碼(func.icd9)的患者並依據居住區碼分群。

doi:10.6342/NTU201601304

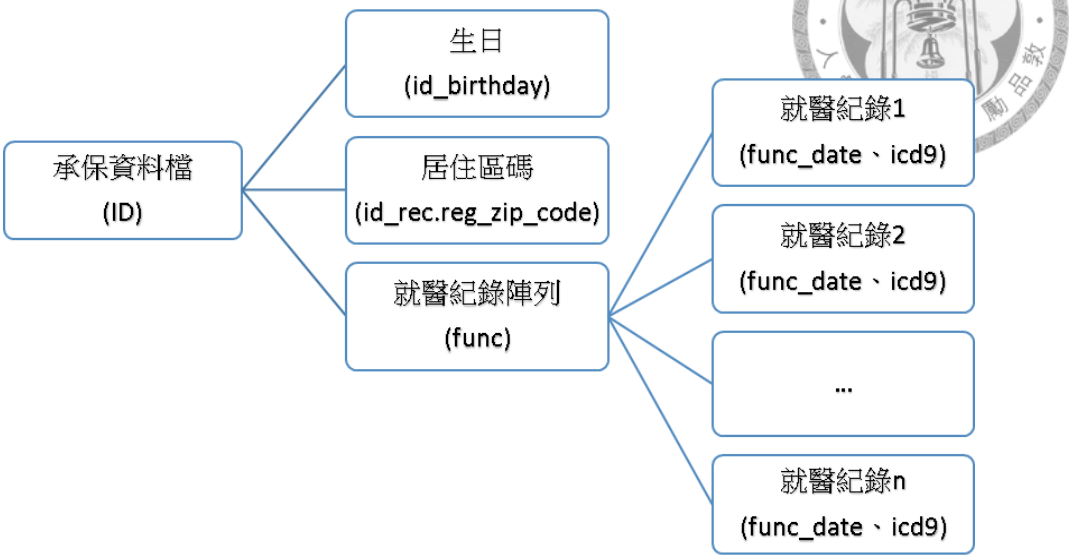


圖 2-1：健保資料庫檔案架構

觀察就醫紀錄，每一位具有健保身分的患者其基本資料(以加密的身分證字號、生日、戶籍)會歸類在承保資料檔(ID)中，而一位患者可能同時擁有多次就醫紀錄，無論門診、住院以及取藥等都會記錄每次的就醫行為於就醫紀錄(func)中，這樣的結構經過長時間累積下來的資料就成為今日所使用的全民健保資料庫。

doi:10.6342/NTU201601304

2-2 水庫水質監測資料庫

行政院環境保護署為配合行政院開放資料 (Open Data) 政策，建置「環境資源資料開放平台(OpenData.epa)」[9]自 102 年起陸續推動將環境資源數據資料彙整開放，藉以提升環境資源資料運用效率落實環境資源資訊共用共享，這些資料集用原始資料(Raw data)的結構提供程式開發者使用。

此研究中也使用環保署環境資源資料開放平台所發布之水庫水質監測資料，欄位包括測站名稱、水庫名稱、所在鄉鎮、所在縣市、測站座標、測項名稱、測項數值以及測項單位，檢測項目包含: 葉綠素 a、總磷、溶氧飽和度、氨氮、化學需氧量、溶氧(電極法)、懸浮固體、氣溫、採樣深度、導電度、濁度、酸鹼值、透明度、水溫、卡爾森指數等指數。選用臺灣本島 17 個縣市共 224 個測站，測站座標數據方便定位於 google map，測站區域代碼方便與健保資料庫做關聯，將測項結果依據採樣時間歸納於其所屬的測站。

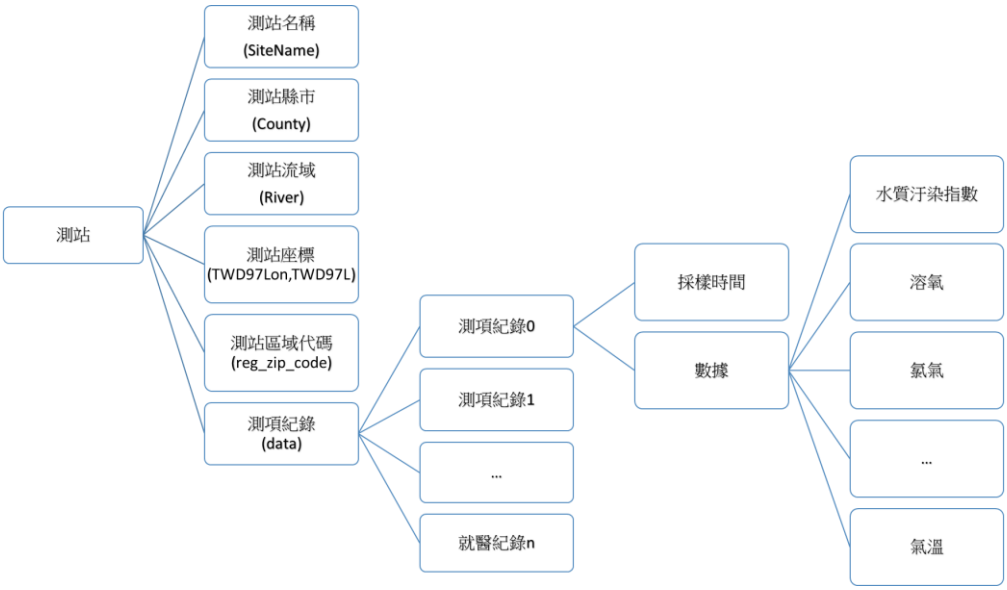


圖 2-2：水質資料庫檔案架構

doi:10.6342/NTU201601304

2-3 自來水水質抽驗資料

由於現代人不會直接飲用河川的水而是取自自來水系統，所以取用行政院環境保護署環境資源資料庫(erdb.epa.gov.tw)[10]自來水水質抽驗資料，屬性包含縣市、自來水分區、測點、採樣日期、氫離子濃度指數、硝酸鹽氮、銀、氟鹽、是否合格、亞氯酸鹽、亞硝酸鹽氮、戴奧辛、氰鹽、汞、溴酸鹽、砷、硒、總三鹵甲烷、鉛、鉻（總鉻）、銻、鎘、鎳、氨氮、氯鹽、硫酸鹽、總溶解固體量、總硬度、自由有效餘氯、酚類、銅、鋅、鉬、錳、鐵、陰離子界面活性劑、"1,1,1-三氯乙烷"、"1,1-二氯乙烯"、"1,2-二氯乙烷"、三氯乙烯、四氯化碳、對-二氯苯、氯乙烯、苯、原水濁度、濁度、臭度、色度、大腸桿菌群密度、總菌落數、"2,4-地"、一品松、丁基拉草、亞素靈、加保扶、大利松、安殺番、巴拉刈、巴拉松、滅必蝨、納乃得、達馬松、靈丹...等重金屬以及有機化合物，由於認為大腸癌僅只與硝酸鹽類與菌落數有關，故僅取用硝酸鹽氮、亞氯酸鹽、亞硝酸鹽氮以及菌落數，原始資料取樣時間從 2001 年至 2015 年的每個月份去除外島資料並依據縣市以及採樣日期之年分依據不同屬性結構化的規劃成資料庫，系統動態載入資料庫時將每年各屬性之所有抽樣數值平均作為其值。

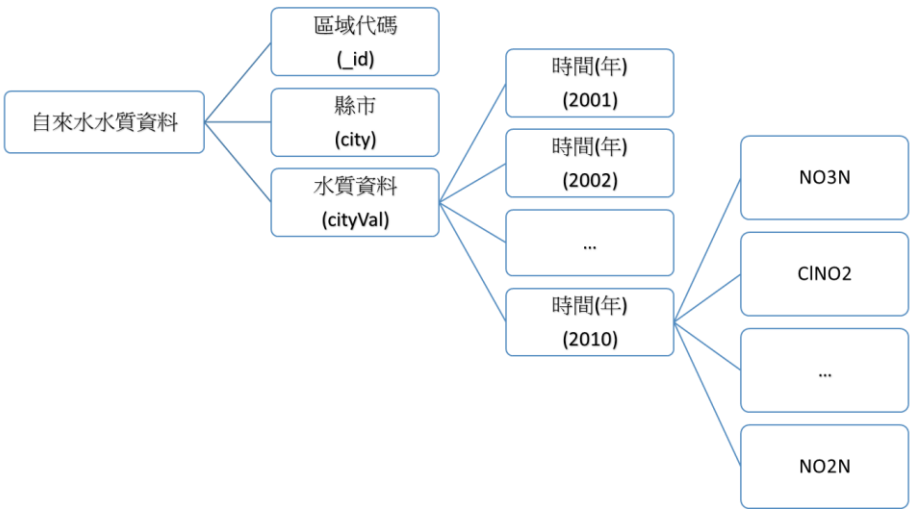


圖 2-3：自來水水質抽驗資料檔案架構

doi:10.6342/NTU201601304

2-4 D3(Data-Driven documents)

欲凸顯健保資料的意義,視覺化成為一個重要的議題,D3 亦稱為 Data-Driven Documents[11],二零一一年被一位史丹福博士生麥克博史達克 Mike Bostock 所創造,將原始資料加以形狀或者顏色呈現不同的分布,利用簡單的折線圖到精緻的動畫讓資料用淺顯易讀並充滿互動的方式呈現,並且能設計系統專屬的呈現模式,如今已成為許多視覺化的設計工具[12]。

D3.js 為視覺化函式庫,使用 SVG (Scalable Vector Graphics)[13]進行繪圖,有別於傳統 HTML 標籤無法精準地定義形狀,SVG 以 XML 標籤格式撰寫,具有可被搜索的特性並具有向量圖型之優點且可以隨時保持圖像清晰不會隨著放大與縮小而失真,也不會增加檔案大小造成存取負擔,從簡單的直線(line)、曲線(path)、圓形(circle)、矩形(rect),到複雜的任意路徑、座標轉換與動畫,均可以利用標籤的形式來完成。

D3 繼承 javascript 使用物件導向的設計模式,如同函式,讓使用者以物件的方式控制、操作並加以應用,也運用了 DOM(Document Object Model)架構作為 SVG 文件的程式介面讓程式可以存取並改變文件架構、風格和內容(例如:attr、element 或者 event)。系統地理圖形相關的演算法輔助讀取地理資訊檔、繪製行政區塊並且做各種投影,利用 D3.js 函式庫所提供的 d3.geo 系列函式進行向量繪圖、座標轉換,建立疾病地圖、疾病趨勢圖以及整合趨勢圖。搭配事件可以做到與使用者互動的介面讓使用者能更貼近資料傳達的意義。

doi:10.6342/NTU201601304

2-5 地理資訊資料

臺灣行政區視覺化最常碰到的問題便是許多地圖系統並不支援臺灣地區，即使有支援通常也不夠完整；近年來由於政府開放資料的推動，臺灣行政區域圖以可以於政府資料開放平台(data.gov.tw)取得，政府釋出了三個等級的資料：縣(市)行政區界線、鄉(鎮、市、區)行政區域界線 [14]以及全國村里界圖，但也由於原始資料提供於政府內部系統採用地理資訊系統檔案格式 SHP（shapefile）一種空間資料開放的向量格式並且採用 EPSG:3826(TWD97/121 分帶)座標系統，但是 D3.js 無法直接讀取 SHP 格式，轉換成 topoJSON 格式方能使用，topoJSON 為 geoJSON 之擴充版本 是一個改善 geoJSON 存取過多的重複資料所做的新的一種資料格式，也因本系統僅需投影臺灣地圖於 SVG 區塊上故採用 topoJSON 為地理資訊檔案格式，topoJSON 文件整體檔案大小也小很多，檔案至少包含幾個部分:type(類別)、objects(物件)、arcs 以及 transform，一個 GeoJSON[15] 物件可以用來代表點 (Point)，線 (LineString)，多邊形 (Polygon) 等等的幾何結構，以及特徵 (Feature) 的集合，或是一系列的特徵 (FeatureCollection)。

1. Type:定義本地理資料的解析方法而且其值必須是 “Topology”, “Point”, “MultiPoint”, “LineString”, “MultiLineString”, “Polygon”, “MultiPolygon” 或 “GeometryCollection” 其中之一，本檔案使用 “Topology” 。
2. Objects:包含多組由使用者定義的地理資料物件集合包含定義物件的類別(type)以及地理資訊(geometries)兩個屬性，本檔案中的物件所使用的類別(type)是”GeometryCollection”， 地理資訊(geometries)中紀錄了各地區的屬性資料，例如:縣市名稱、鄉鎮名稱、區域面積以及地區代碼…等，以及邊界點座標。
3. Arcs:負責記錄較常被取用的拓樸繪圖點座標。

doi:10.6342/NTU201601304

4. Transform:中設定適合的縮放比例以及偏移量在初始化時有適當的視覺效果而不需再投影時設定其參數。

本地理資料集仍然有許多部分需要處理，例如常見的異體字問題需要統一用字，2010 年臺灣實施五都改制施行縣市合併、升格直轄市，除了行政區名稱的改變外也影響區域代碼，為了便於系統設計所以採用新制，其次，臺澎金馬涵蓋的範圍相當廣，若要將所有區域納入同一份地圖，會有許多空間消耗在島嶼與島嶼之間的領海中，也因為本系統所使用的資料僅限於臺灣本島排除資料較少的外島。

doi:10.6342/NTU201601304

第三章　系統設計

本章節主要介紹系統架構與技術，系統主要分成資料庫伺服器端與網站伺服器端，分別負責運算以及顯示兩個部分，由網站伺服器負責與使用者互動以及資料呈現，資料庫伺服器端負責資料的分散式運算與資料儲存。

3-1 系統環境

網站伺服器 Web Server 與資料庫伺服器 Database Server，兩系統硬體配置如表 3-1 所示。

	網站伺服器 Web Server	資料庫伺服器 Database Server		
作業系統	Windows 7 Enterprise SP1	Ubuntu 12.04.4 LTS		
處理器	Intel® Core™ i5-2400 Processor 3.40 GHz	Intel® Core™ 2 Quad Processor 2.40GHz	Intel® Core™ i5 CPU 2.67GHz	AMD FX™-8350 Eight-Core Processor 1.4GHz
記憶體	16 GB	8GB	16GB	32GB
系統類型	64 bits	64 bits	64 bits	64 bits

表 3-1：系統環境

doi:10.6342/NTU201601304

3-2 NoSQL

大數據資料格式多元，所以結構化的資料庫在擴展及應用上容易受到限制，典型的關連式資料庫在密集型資料的應用上都有效能欠佳的問題，包括索引大量的文件，對於高流量的網站容易造成效能負擔，對於串流媒體格式無法提供存取，所以採用 NoSQL 資料庫來提升效能與擴充彈性。

NoSQL 的興起有鑑於網路應用越來越廣泛，除了使用電腦外更多人使用行動裝置存取網際網路，導致多元資料快速增加，傳統關聯式資料庫的運算策略並不適合處理大量資料，而需要一套分散式資料庫系統負責處理以及運算。早在 NoSQL 資料庫這個名詞流行之前，就已經出現了很多種非關聯式資料庫，這些資料庫各有不同的特徵，目前有 4 種比較受到關注的 NoSQL 資料庫，分別是 Key-Value 資料庫，記憶體資料庫（In-memory Database）、圖學資料庫（Graph Database）以及文件資料庫（Document Database）。資料通常採用本地儲存的方式，可以透過增加節點的方式來擴充容量，NoSQL 適用於大資料的處理，有別傳統關聯性資料庫的強一致性，NoSQL 強調最終一致性，因此不支援增加運算複雜度的 join 指令，並不再每筆運算進行時就將運算結果寫入資料庫，而是希望無論執行狀況如何都能讓運算持續下去，最終會導向正確的結果，但背後的風險其實就是資料遺失。

doi:10.6342/NTU201601304

3-3 MongoDB

mongoDB 是一種 document-based NoSQL 資料庫系統，資料體結構是以鍵值 Key-Value 組合，採用 JSON 格式儲存資料，有類似關聯式資料庫表格的資料結構，但 Schema-less 特性不需預先定義 schema，對於資料的擴充能力有很大的幫助，可以處理 Terabyte 量級的資料也就是大數據的資料庫，MongoDB 並不支援 join 語法所以如果欲在多個 collection 中搜尋只能透過多次搜尋，所以再建構資料模型時也必須做好規劃，以減少效能耗損。

為了解決資料量超過單一機器儲存上限的問題，所以使用水平擴張的模式建立叢集系統，分片(Sharding)是將儲存資料跨不同機器的方法，每一個 Shard 可以視為一個獨立資料庫分散彼此的資料同時也互相備份，Shard Key 可以用於資料切割，也可以用來提高 MongoDB 查詢資料的效率，資料分散在不同的 Shard 上，當使用者要查詢資料時，一種作法是對每一個 Shard 都執行查詢指令，最後再彙整出查詢結果。另一種作法是查詢指令也包含 Shard Key 鍵值，直接到 Shard Key 所對應的 Shard 中查詢，而不需要對每一個 Shard 執行這個查詢指令，可以提高查詢效率；分片可以在主機間進行複製，並且至少由一部電腦來管理；MongoDB 採用非同步複製機制來獲得更高的效能。MongoDB 叢集以三台電腦作為一個基本單位，各自扮演不同角色包含：Shard Server、Config Server 以及 Route Server，Shard Server 用於儲存實際數據，實際生產環境中一個 shard server 角色可由幾台機器組個一個 replica set，防止單一節點故障，Config server 存儲了整個 Cluster Metadata，其中包含 chunk 信息，Route Server 作為用戶端接入的前端路由，讓整個叢集看起來像單一資料庫方便前端應用。

doi:10.6342/NTU201601304

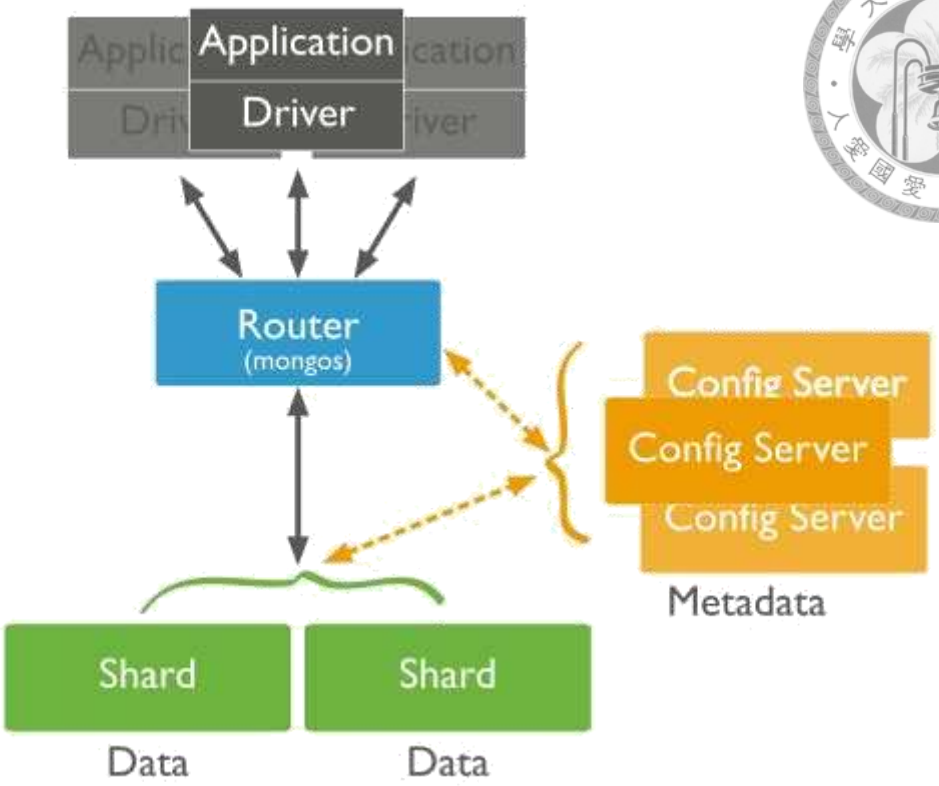


圖 3-1：MongoDB 叢集

doi:10.6342/NTU201601304

3-4 D3(Data-Driven documents)

雖然有許多視覺化工具支援地圖視覺化，但在臺灣卻往往看不到比縣市還更小層級的區塊，利用開源軟體成為常見的解決方案之一。

1. 疾病地圖:

系統利用政府所提供的 shapefile(.shp)檔案並進行前處理將之轉為 topoJSON 格式之 sector.js 以及 county.js 分別具有不同層級的行政資料，前者為縣市後者為鄉鎮，每個行政區塊均會對應到一個 properties 物件，利用 topoJSON 模組提供的函式讀取該檔案並搭配 D3.js 設計疾病地圖以及疾病趨勢圖，疾病地圖使用 D3.js 函式庫中的座標轉換 d3.geo.path 搭配 d3.geo.mercator 來繪圖將地理資訊轉換成 SVG Path 標籤形式的臺灣地圖物件投影於該 SVG 區塊；利用迴圈便能將該地區對應的數值以及疾病盛行率以著色以及屬性(attr)的形式附加於對應的行政區塊物件中，搭配事件讓疾病地圖具有互動的功能，系統初始化時地圖著色帶有透明度，當使用者將游標移入或者選取地圖中的縣市，該區域邊界會用較粗的黑色實線來標記，便於使用者判讀地圖中的區域是否選取。

建立疾病分布比例尺作為系統著色的依據，根據疾病盛行率便可以作為各區塊著色的輸出函式，利用線性分布函式 d3.scale.linear().domain().range()，如圖 3-2 所示，domain 代表其輸入所涵蓋的範圍，range 代表其輸出的範圍，本系統將此變數依據疾病發生率數值由高、中至低分別設定為紅(#E22),黃(#EE0),綠(#0D0)三個主要級距以達到較好的輸出視覺，可以依據疾病盛行率投影對應的顏色，在地圖旁宣告與縣市數等量的矩形，填入對應的顏色並註冊事件，讓地圖與比例尺能夠互動，切換時間時疾病分布比例尺也會進行更新。具有相對盛行率與絕對盛行率兩個模式。

doi:10.6342/NTU201601304

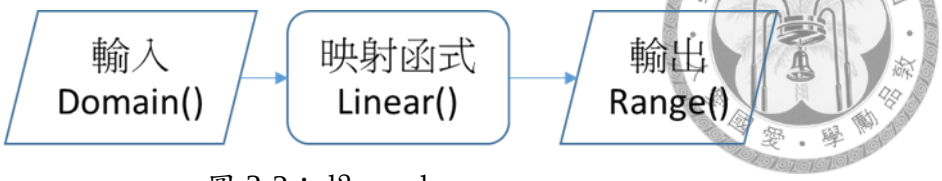


圖 3-2：d3.scale

2. 疾病趨勢圖：

利用分頁功能，讓介面更加簡潔，包含三個功能：地區疾病趨勢、疾病趨勢比較以及基本表格。

A. 地區疾病趨勢:

此介面透過滑鼠游標移入疾病地圖之任意縣市時更新，系統計算各年度之數據：疾病通報數、平均年齡(各年齡層分佈)以及疾病盛行率與前一年之差異值分別呈現於三組折線圖中，水平座標為以年為單位之時間軸，垂直座標為數值，當使用者將滑鼠移入任意折線圖時，系統會抓取該資料點所代表的數據，以該時間資料為依據，掃描其他兩張圖之資料點，進行標記並顯示出數值以達到同步的效果；使用者也可以透過點擊疾病趨勢圖中的資料點，疾病地圖會根據該資料點的所代表的時間進行同步。

B. 疾病趨勢比較:

主要作為不同地區之數值趨勢比較功能，補足一疾病趨勢圖僅能顯示單一區域的不足，使用者從疾病地圖中選擇欲比較之縣市，系統會將狀態儲存於陣列中並繪製折線圖，可以選擇疾病盛行率、通報數以及人口...等屬性，為了提升可讀性該年度會被系統標記，便於比較該年度之疾病數值。

doi:10.6342/NTU201601304

C. 基本表格:

此模組將疾病地圖以表格方式呈現該年度的疾病趨勢並且會隨著使用者操作而更新，每一列包含色標、縣市名稱、疾病盛行率、疾病通報數以及平均年齡，會根據疾病盛行率進行排列，系統會計算並顯示與去年度之差值便可以瞭解該縣市之疾病趨勢，會依據時間自動更新並可以與疾病地圖進行互動，當使用者使用游標移入疾病地圖中的縣市時，表格中所對應的資料列便會被標示。

3. 整合趨勢圖:

此介面希望能將疾病之盛行狀況與水質資料整合，傳統折線圖僅呈現多組資料於二維座標，若欲加入環境資料作為第三個屬性，選用立體的三維座標會造成畫面排版凌亂不易閱讀，所以使用泡泡圖[16]加入半徑作為第三象限，水平座標為疾病相關數值，使用者可以選擇疾病盛行率與通報數，垂直座標為水質資料庫數據屬性，半徑為疾病通報數，繪製路徑以觀察環境資料與健保資料的趨勢，由於希望使用者能夠更貼近數據趨勢所以透過調整時間或者利用拉動事件透過滑鼠在該縣市的路徑上拖曳，系統便會依據其半徑(radius)顯示縣市之名稱。

doi:10.6342/NTU201601304

3-5 cache 系統

系統在執行資料搜尋時使用 mapreduce 作為在分散式系統中抓取資料的技術，其優點是利用 map 函式將查詢分散為更小的程序以達到分散的效果，最後利用 reduce 函式彙整出完整的結果，但如果每次執行查詢皆須透過資料伺服器抓取資料會造成系統的負擔也會導致系統查詢沒有效率，所以建立 cache 系統以減少對資料庫的存取，當多位使用者同時使用時也會有較好的效率。

從資料庫伺服器中存取資料，對於較重大疾病例：如呼吸道疾病(ICD-9：480、481、482、483、484、485、486、487)以及腸病毒(ICD-9：047、048、049、074)，在執行效能上會造成重大負擔，所以建立 cache 如圖 5-1 所示以減少對於資料庫伺服器的存取機會，若有多位使用者同時操作時也不會造成系統的負擔，使用系統時先輸入欲查詢之疾病代碼(ICD-9)、藥物(drug)以及是否為傳染病(infectious)等條件，系統便會從「快取映射資料庫」中依照查詢條件檢查快取中是否有對應的紀錄 collection，如果存在對應的 collection，系統便可以直接從「快取資料庫」中讀取搜尋結果並輸出；如果不存在對應的 collection，系統便會從資料庫伺服器中利用 mapreduce 存取對應的病患資料。

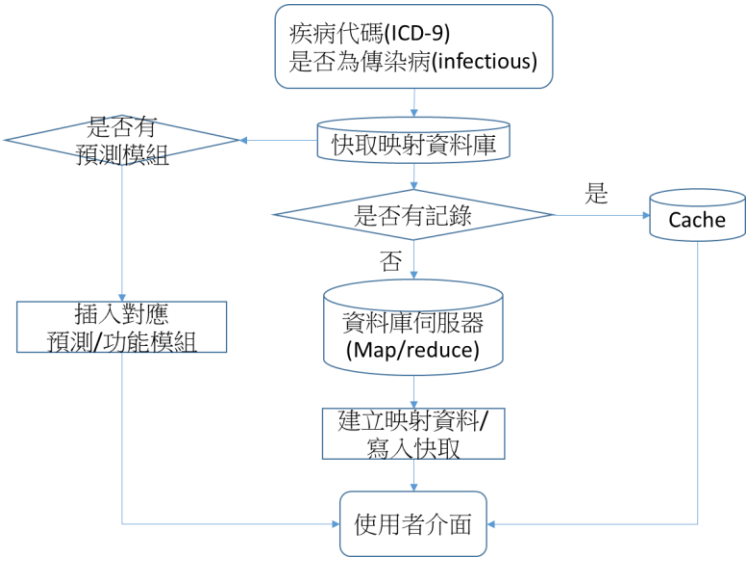
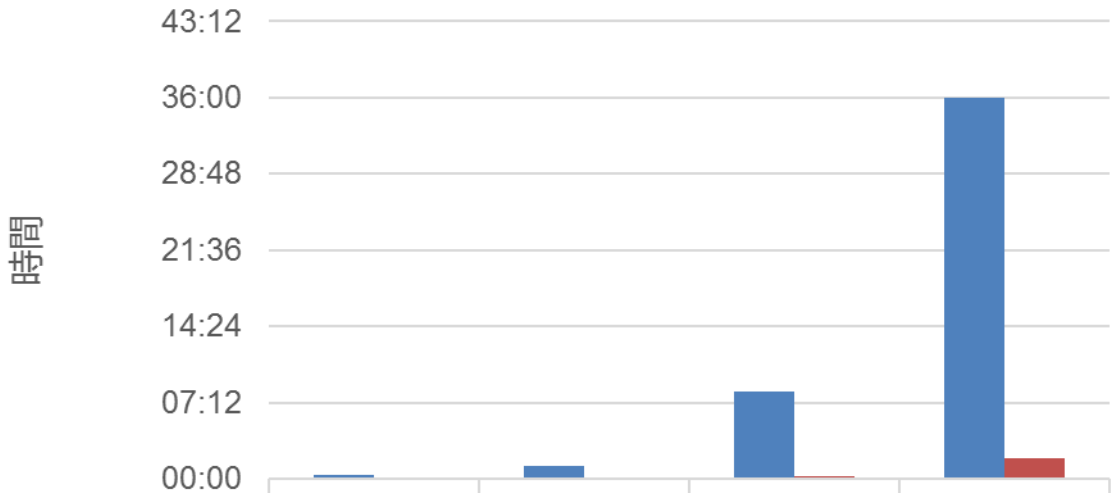


圖 3-3：快取存取流程

doi:10.6342/NTU201601304

測試之疾病		
疾病代碼	名稱	病患人數(人)
061、0654	登革熱	971
1540~1543、1548	大腸癌	1663
250~252	糖尿病	7001
480~488	呼吸道疾病	33168

表 3-1：測試之疾病代碼



	登革熱	大腸癌	糖尿病	呼吸道疾病
資料庫存取時間	00:00:22	00:01:10	00:08:16	00:35:55
快取存取時間	00:00:03	00:00:03	00:00:12	00:01:53

表 3-2：測試結果

doi:10.6342/NTU201601304

3-6 系統架構

MVC 架構將網站伺服器應用程式中的資料模組(Model)、呈現模組(View)、控制模組(Controller)三者獨立出來如圖 3-2 所示，控制模組(Controller)是所有指令的核心，資料模組(Model)負責資料庫以及邏輯，呈現模組(View)負責客戶端視覺化的部屬相關，將系統模組化後對於系統的開發與維護有很大的幫助。

本系統採用動態網頁應用程式做為開發基礎，為了讓系統開發具有彈性，選用 Codeigniter 這套 framework 架設 MVC 架構為基礎的網頁應用程式，使用者從網站伺服器載入呈現模組(View)入口網站中的的表單填入並將請求交由控制模組(Controller)進行數據前處理，根據使用者輸入將需求交給資料模組(Model)使用 MapReduce 技術從資料庫伺服器中 MongoDB 的分散式資料庫依據查詢條件讀取需要的資料，解析其結構並回傳給使用者，在資料庫伺服其與網站伺服器之間利用 cache 系統儲存查詢紀錄以提升查詢效能，系統會根據使用者的查詢條件置入特定的呈現模組。

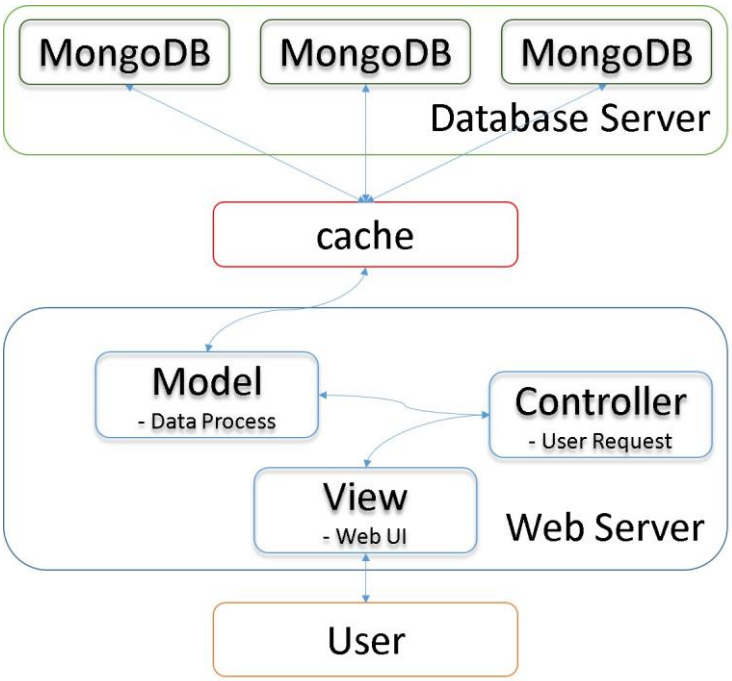


圖 3-2：MVC 架構

doi:10.6342/NTU201601304

第四章　研究方法

本章將介紹資料前處理、資料庫建置、分散式資料庫技術到最後本系統視覺化工具所應用的所有方法；資料前處理將資料彙整成系統所需之資料結構，儲存於 MongoDB 分片資料庫伺服器中，並運用 MapReduce 功能存取資料，最後透過本系統設計之視覺化工具呈現資料，方便使用者閱讀資料。

4-1 資料預處理

原始數據(raw data)沒有結構且具有許多冗餘欄位，可能隱藏著許多問題甚至可能影響效能，必須透過一定的預處理程序將數據結構化為系統容易使用結構，並建構成鍵值對儲存於資料庫中。

4-1-1 水庫水質監測資料庫

原始數據包含兩組檔案，水質測站基本資料與水質採樣數據，兩個資料可以透過測站的名稱進行關聯。

水質測站基本資料，包含測站名稱、所在縣市、區域代碼、鄉鎮、測站流域以及座標，方便將資料視覺化整合於地圖上，由於僅選取本島做為參考資料所以剔除非本島亦即位於離島(澎湖、金門、馬祖…等)的測站。

水質採樣數據的預處理，先將原始數據以測站所在區碼(zip_code)歸檔，在依據採樣日期從 2001 年到 2010 年的每個月分、採樣名稱以及採樣數據進行歸類，建立巢狀結構資料庫。

4-1-2 自來水水質抽驗資料

由於原始資料並不包含鄉鎮僅具有測點這個屬性，故須從測點透過縣市將鄉

doi:10.6342/NTU201601304

鎮解析出來並將其與區域代碼關聯，剔除外島並依縣市改制後的行政區歸檔，將較常使用的採樣屬性（如:NO3N、NO2N、ClNO2…等）依據採樣日期之年分歸檔。

4-1-3 健保資料庫

健保資料庫中的資料每一筆紀錄結果皆為歸人的結構，包含經過轉碼的身分證字號、生日、性別、戶籍區碼以及其就醫紀錄。健保資料庫將資料透過水平擴充技術分片(Sharding)分散儲存於三台 mongoDB 資料庫伺服器中，存取時透過 mapreduce 技術解析資料提升效能。

4-2 分片(Sharding)

分片(Sharding)是 MongoDB 將資料儲存在叢集的一種方法，用於分散單一伺服器負載就是將資料庫分區塊放在不同的資料庫節點上。大量的針對單一個 database daemon 進行 query，會造成 CPU 的運算負擔。大量的 data sets 也可能會超過單一台機器所能提供的硬碟容量，所以利用分片(shard)來突破硬體限制。

4-2-1 分片目的

傳統資料庫架構是為了能在單一主機上運作順暢，但若單一機器需要承擔大量資料以及高度的運算量對將會成為負擔，大量的存取可能造成中央處理單元(CPU)的效能用盡資料量過大連記憶體不足以應付甚至超出伺服器硬碟儲存的上限。

面對硬體結構上的效能限制，大型系統提高效能的方法有兩種，垂直擴充(vertical scalability)與水平擴充(horizontal scalability)。最簡單的方法就是垂直擴充，透過提升中央處理器(CPU)、擴充記憶體(RAM)以及增加硬體容量，以最直接的方式提升效能加速資料庫的處理效能，然而這樣的擴充方法仍然會造成硬體上

doi:10.6342/NTU201601304

的限制，擴充效能相較於水平擴充其實並不符合成本。水平擴充，有效利用多台相對低階的資料庫伺服器分散工作負荷以提升效能和可靠性達到甚至超過一台較高階的伺服器主機，處理分散式系統具有一定的困難性需要考量速度、擴充性、容錯性和一致性；MongoDB 中水平擴充方法又稱為分片(sharding)， MongoDB 資料庫的自動分片技術就是將原先資料庫中集合依據一定的規則切成若干分片(shards)分散於叢集中，這些分片小塊可以視為獨立資料庫統一由 mongos 路由管理以使用者觀點組成一個完整的資料庫，當有請求查詢或寫入時，路由會依據分片 shard key 規則找到對應的分片進行操作。

4-2-2 MongoDB 分片機制

一個 MongoDB 的 Shared 叢集由三個角色組成:

1. 分片(Shards):

   用來儲存分片後的資料，通常是一組副本集(replica set)用來提高可用性和資料一致性。

2. 查詢路由(Query Router):

   在 MongoDB 裡稱為 mongos 是讓用戶端能直接操作對應 shard 的介面，可以針對相關的片段進行存取。

3. 配置伺服器(Config Server):

   紀錄叢集的 metadata 以及資料與分片對應的資訊。

欲對一個 collection 作分片必須選擇 shared key，MongoDB 再根據 shared key 的值將資料分割成許多個 chunks 並平均分布在 shard 中。針對 shard key 的儲存策略分成兩種，分別是 range based 與 hash based 兩種，前者直接判斷鍵值將集合切割成多個 chunks 而後者則會透過雜湊函式(hash function)來分配其屬於之 chunk；為了維持分散資料的平衡，MongoDB 使用 splitting 和 balancer 兩個背景程式來完成，當 chunk 成長為設定的大小時 Splitting 這支程式會將 chunk

doi:10.6342/NTU201601304

分成兩半，確保 chunks 的成長不會太大；Balancer 來管理 chunks 資料的搬移，但這樣過程中，會產生許多資料移動的成本，要盡可能的讓這行為發生的次數降低，當最多 chunk 數的 shard 比最少 chunk 數的 shard 多 9 個 chunk 時 MongoDB 才會開始做 Balancing 移動、合併資料的動作。

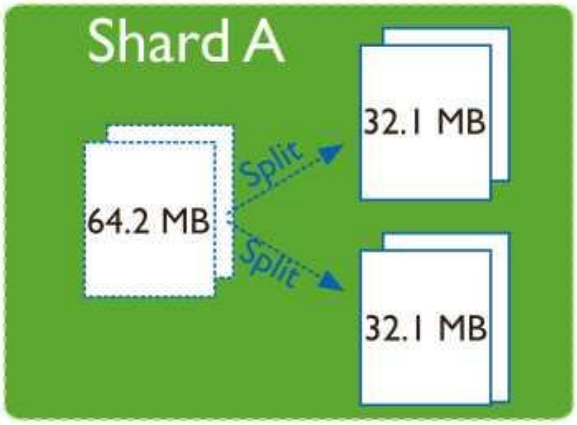


圖 4-1：Splitting

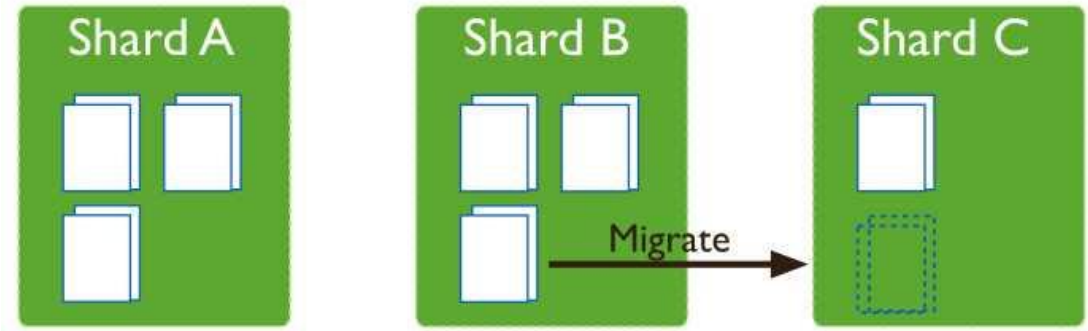


圖 4-2：Balancer

doi:10.6342/NTU201601304

4-3 MapReduce

MapReduce 是 MongoDB 內建分析資料的介面，讓使用者可以從分散式資料庫中取得資料並提高查詢效能並能進行複雜的運算。

其運作機制，map task 會從 input shard 中讀取指派給他執行的資料，並將對應需要處理的資料透過許多 map worker 同時執行，提升 performance，map worker 執行完 map function 內的程式後，最後的 output 仍是(key,value)這個資料串，這些中間資料會先暫存成 spill 檔在記憶體中，最後會將所有 spill 檔進行合併(merge)與排序(sort)，目的是把所有有相同 key 值的資料聚集在一起，當所有 map worker 完成工作時，Master node 就會通知 Reduce workers 執行 reduce function，每讀入一個新的 key 值時就會呼叫一次 reduce function 並傳入 key 值以及相同 key 的 value 所組成的 list，執行完後結果會寫入記憶體，當所有 reduce worker 完成工作，Master 會將控制權轉回 user program。

使用者從入口網站填寫欲查詢疾病之疾病代碼以及其他條件，如圖 4-3 所示，每次資料搜尋會根據使用者輸入欲查詢的疾病代碼(ICD-9)，系統存取資料庫運用 MapReduce 從龐大的歸人文件資料庫中取得具有特定就醫紀錄的病患資料並將不符合條件的紀錄去除，透過 Reduce 程式將資料利用戶籍區碼進行歸檔，並會依據輸入的疾病代碼建立快取，未來若有相同的查詢無須經過重複對資料庫伺服器存取的 Mapreduce 動作而是直接從已有的快取中取得結果;「是否為傳染病」的選項會影響統計的方式，由於傳染病(腸病毒、流行性感冒…等)的治療週期較短故在統計時會將每月的就醫紀錄皆會列入統計，而非傳染病(癌症)的治療周期較長，故在統計時若曾有過就醫紀錄便不再進行統計。

系統中疾病盛行率的計算方式為疾病通報數除以縣市總人數，縣市總人數為系統建置前由健保資料庫統計而成，由於健保資料不包含人口數值資料，為了往後欲計算疾病盛行率所需要的各縣市人口數，所以在建立資料庫時直接從百萬抽

doi:10.6342/NTU201601304

樣資料中掃描每一位病患並將其依照居住區碼統計而成。

圖 4-3：入口網站

doi:10.6342/NTU201601304

4-4 資料呈現

資料呈現是數據視覺化的核心也是本系統的主軸，透過系統操作以及圖表將疾病資料庫欲傳達的訊息整合並將趨勢呈獻給使用者，系統將視覺化分成空間與時間兩個面向，空間面向以疾病地圖的方式呈現，時間面向則是以疾病趨勢圖來呈現以及整合趨勢圖，使用者可以利用控制面板調整欲查看的時間點或是利用展示功能觀察不同年份間的疾病趨勢。

4-4-1 疾病地圖

疾病地圖為空間面向的資料呈現，在視覺上呈現出同一個年度中臺灣各縣市的疾病盛行率分布狀況，此介面包含兩個部分：疾病地圖以及疾病分布比例尺。

疾病分布比例尺作為著色依據，利用利用 D3.js 的 scale.linear()函式，包含 domain()與 range()兩個屬性分別為輸入範圍與輸出範圍，輸入範圍會依據相對盛行率與絕對盛行率兩個模式而有所不同，相對盛行率模式中系統會計算該年度疾病發生率的極值，所以在疾病地圖上會有較明顯的顏色對比，絕對盛行率模式中系統會計算所有年度中疾病發生率的極值，當使用者在年度間切換時可以根據顏色變化的趨勢理解該疾病在臺灣發生的趨勢。輸出範圍由數值高至低分別設定為紅色(#E00)、黃色(#EE0)與綠色(#0E0)三種顏色；輸入範圍若僅設定為極大值、極大值與極小值的平均與極小值在絕對盛行率模式下會造成顯示上不明顯，以登革熱為例 2002 年席捲南臺灣也因此導致疾病盛行率極大值與極小值的平均值因為 2002 年而偏高,從疾病地圖上可以看出除了該年度高雄因為疫情嚴重而以紅色著色，其他區域之著色則都是深綠，所以修正極大值與極小值的平均值為疾病發生率的平均值以增加不同區域的對比度而不會因為某個極端質而影響整體呈現。

此外，系統為了提升使用者對於疾病在分布的使用者經驗，透過互動事件讓

doi:10.6342/NTU201601304

使用者操作疾病地圖:

1. 滑鼠移入:

利用滑鼠移入地圖中縣市，便會有漂浮的區塊顯示該地區的詳細資訊，包含年度疾病通報數、人口以及年度疾病盛行率，並標記對應的疾病比例尺，疾病趨勢圖便會立刻繪製該區之疾病通報數、盛行率、年齡分布以及該縣市中所有鄉鎮之疾病分布狀況。同時也可以透過疾病比例尺依據不同顏色所代表的疾病發生率選取特定疾病發生率的地區代表。

2. 滑鼠點擊:

點擊疾病地圖中的縣市即觸發選取事件，該縣市的邊界會被粗體標記，系統會記錄使用者的選取之縣市並在疾病趨勢比較圖中繪出不同地區的疾病數值方便使用者閱讀並比較不同區域中疾病通報數、人口以及疾病發生率的變化趨勢。

3. 地圖縮放:

利用 SVG(可縮放向量圖形，Scalable Vector Graphics)的物件特性，不會隨放大與縮小而有所失真可以保持圖像清晰度，將 SVG 物件註冊縮放事件，讓使用者可以透過滑鼠滾輪對地圖進行縮放以及拖曳，利用 d3 資料庫的 behavior.zoom()函式進行縮放，scaleExtent 屬性設定縮放級距的最大與最小值系統設定為[1,10]，如此便能方便使用者選取欲察看之地區。

4-4-2 疾病趨勢圖

疾病趨勢圖為時間面向資料呈現包含三個功能：地區疾病趨勢、不同地區疾病趨勢比較以及基本表格，為了避免系統畫面中一次傳達過多資訊所以利用分業的方式模組化，讓系統一次展示一個功能，並利用事件在最適合的時間點互動，利用折線圖呈現出各年度之疾病通報數、人口以及疾病盛行率。

doi:10.6342/NTU201601304

1. 地區疾病趨勢圖:

滑鼠移入地圖中縣市，疾病趨勢便會立刻繪製該區之疾病詳細數據以及該縣市中所有鄉鎮之疾病分布狀況，包含年度疾病通報率、年度人口以及年度疾病盛行率三個趨勢圖並且計算每年度與前年度之差值，其中趨勢為時間相關資料所以使用折線圖來呈現，差值則選用長條圖讓使用者可以清楚地看出差值是否增加或者減少，系統將三張趨勢圖進行同步，使用者可以利用滑鼠移入任意趨勢圖中的資料點，系統便會掃描其他趨勢圖並且標記顯示出該時間點之數據以達到同步的功能，使用者可以透過點擊任意一張趨勢圖中的資料點，疾病地圖便會同時抓取該年度的疾病趨勢並進行同步更新；介面包含鄉鎮疾病分布圖讓使用者能用更細微的視野觀察鄉鎮的疾病分佈。

2. 疾病趨勢比較圖:

點擊疾病地圖中的縣市時觸發，系統會記錄使用者的選取的縣市，依據所選擇的屬性，包含:通報數、平均年齡、盛行率、18 歲以下疾病盛行率、19 至 30 歲疾病盛行率、31 至 44 歲疾病盛行率、45 至 64 疾病盛行率、65 歲以上疾病盛行率，抓取對應的數據組成較小的集合，並在疾病趨勢圖中繪出不同地區的疾病數值以比較不同區域的趨勢差異。

3. 基本表格:

此介面為輔佐疾病地圖的另一種呈現方式，將疾病地圖的資訊利用疾病發生率進行排序以表格的方式呈現，使用者利用滑鼠移入地圖中的縣市，系統便會在表格中標示出該區域的數值，系統同時會計算各數值的年度差異並顯示於表格中利用紅色顯示為增加綠色顯示為減少，用對比的方式讓使用者可以以另一種角度閱讀疾病地圖。

doi:10.6342/NTU201601304

4-4-3 整合趨勢圖

利用 D3.js 建立專屬的整合趨勢圖系統;在 HTML 標籤中宣告一個 SVG 物件並先預留圖表空間，接著利用 D3.js 所提供的 axis API 建立 X 軸與 Y 軸，包含 scale()、orient()與 ticks()三個屬性，scale()定義數值的輸入與輸出，scale()包含三個屬性 linear()、domain()與 range()，linear()定義數值輸入以線性方式映射，有指數(exp)與對數(log)兩種依據輸入數值的結構選擇最適當的映射模式，domain()定義輸入的範圍，range()定義輸出的範圍，需要注意的是;orient()定義軸線所排列的位置與方向，水平 X 軸設定為底部(bottom)在垂直 Y 軸設定為左側(left)；本系統將 X 軸設定為疾病通報數或者疾病盛行率，Y 軸則設定為水質資料(包含 NO3N、NO2N 與菌落數…等)，根據使用者選擇的對照組地區、比較組地區與水質屬性，再建立 X 軸與 Y 軸後便可以輸入資料點，資料點以 SVG 圓形物件(circle)呈現，包含 x 座標(cx)、y 座標(cy)、半徑(r)、著色(fill)與外框(stroke)，利用 D3.js 將物件 append 在趨勢圖上，系統規劃 x 座標為疾病盛行率透過 xScale 函式映射之數值，y 座標為水質資料庫透過 yScale 函式映射之數值，半徑 radius 為疾病通報數透過 radius 函式映射之數值。

初次載入頁面時先初始化，利用 map 函式將資料點對應的疾病資料以及水質取出並利用 position 函式映射出對應的 X、Y 座標點以及半徑，往後時間調整時只需呼叫座標半徑之更新函式(position)即可更新圖表示的圓形物件，當使用者選取不同地點以及水質資料時系統會掃描所選地區以及水質資料庫數值的最大值與最小值，更新 xScale 與 yScale 的 domain()參數並繪製更新後的 xAxis 與 yAxis，讓資料以最適合的方式呈現於趨勢圖中。

每個圓形物件皆註冊事件讓使用者可以互動，當使用者將滑鼠移入圓形物件上時繪出該物件之代表縣市所經過的路徑，透過拖曳事件讓使用者可以將圓形物件在該縣市所經固的路徑上拖曳，同時其他圓形物件也會被掃描並更新 x 值、y

doi:10.6342/NTU201601304

值與半徑。

水質資料庫採用動態載入的模式目的是提升效能，可以透過控制面板的其他資料庫中勾選欲載入的資料庫(目前有自來水水質資料與水庫水質抽樣資料)，系統便會利用 Ajax 從 mongoDB 中載入對應的資料庫，由於不同資料庫均有各自的資料結構及屬性所以必須先進行處理，將資料整合成依據縣市以及年份規劃的通用資料結構，紀錄可用的資料庫屬性更新 Y 軸所能使用的選項，並且將水質整合趨勢圖重新初始化。

doi:10.6342/NTU201601304

第五章　資料呈現

本系統為疾病資料視覺化系統，所以建立一套查詢系統方便使用者針對不同疾病代碼抓取資料，視覺化是本章節主要核心，由於資料種類繁多所以想要以時間與空間等，不同的角度呈現資料的樣貌，減少視覺上的雜訊以精確地傳達數值的意義。

5-1 系統簡介

使用者從首頁輸入欲察看之疾病代碼 ICD-9，以及欲載入之其他資料庫，系統便會從健保資料庫或是快取中讀取資料，完成搜尋後便會進入結果呈現的界面，包含控制面板、疾病地圖、整合趨勢圖。

5-2 首頁

可以自行輸入疾病代碼(ICD-9)或者從選單中選取預先儲存的疾病代碼組合，為了方便一般使用者操作所以建立這組轉譯系統即使不懂疾病代碼(ICD-9)也能輕鬆操作，亦可以勾選其他資料庫在整合趨勢圖中可以使用，送出後系統便會從快取或者資料庫抓取疾病資料進行結構化後，傳入結果呈現的界面。

doi:10.6342/NTU201601304

圖 5-1：搜尋介面

5-3 結果呈現

由於系統的疾病地圖、疾病趨勢圖以及整合趨勢圖三個模組畫面較為龐大不適合呈現在一個畫面中，所以將畫面切為上下兩個部分，如圖 4-4 所示，疾病地圖、疾病趨勢圖做與整合水質資料趨勢圖分開，為了方便使用者操作，所以在畫面左側建立控制面板。

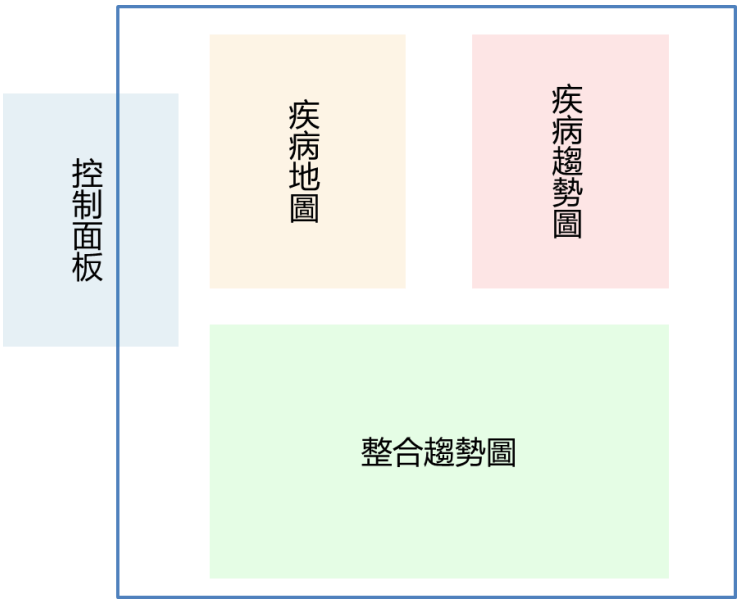


圖 5-2：系統地圖

5-3-1 控制面板

為了便於操作疾病地圖與疾病趨勢圖，所以設計此控制面板，如圖 5-3 所示，具有時間控制與快速移動的錨點功能，時間控制包含：自動展示、年份增加以及年份減少，讓使用者可以調整時間；透過錨點可以快速於疾病地圖與整合趨勢圖中快速移動；此外控制面板顯示此次搜尋的基本資訊，包含疾病代碼以及是否為傳染病以外，使用者也可以利用此介面動態載入其他資料庫，包含水庫水質資料庫以及自來水水質資料庫。

doi:10.6342/NTU201601304

圖 5-3：控制面板

5-3-2 疾病地圖

疾病地圖是一個利用顏色來凸顯疾病的盛行狀態的工具，呈現了同時間不同地區的疾病分布狀況，利用控制面板調整時間或者自動展示功能改變疾病地圖的著色,根據顏色的轉換可以一目了然地觀察疾病在臺灣的盛行趨勢以及分布狀況。

可以選擇相對盛行率以及絕對盛行率兩種著色模式，如表 5-1 所示，在相對盛行率模式中由於僅以該年度作為疾病分布比例尺的極大與極小值，通常會有較強烈的顏色對比，此模式下的紅色僅能代表該疾病之盛行率相對高於其他縣市有凸顯某區疾病盛行狀況的功能，在絕對盛行率模式下地圖中的顏色對比度雖然相對不明顯但看得出年度與年度間整體的疾病趨勢，以大腸癌為例，如表 5-2 所示，隨著時間(年度)增加，疾病地圖的顏色漸漸從綠色調轉變為黃色甚至是紅色，由此可知該疾病盛行率有顯著增加的趨勢，此外也可以利用色標來尋找其對應的縣市，增加使用者的互動性。

doi:10.6342/NTU201601304

相對盛行率	絕對盛行率

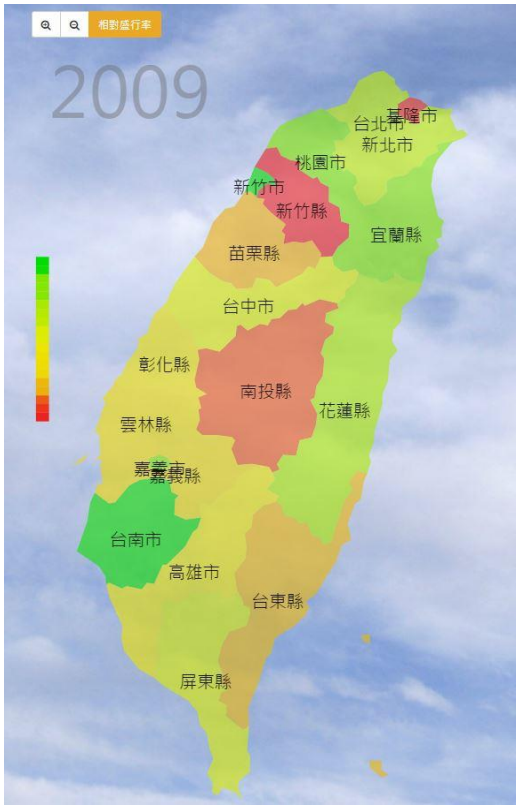
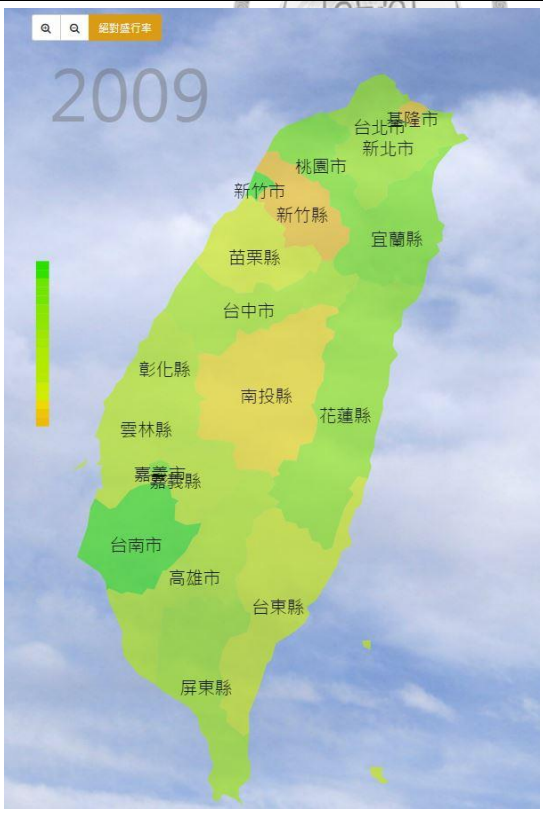

	

圖 5-3：疾病地圖相對盛行率	圖 5-4：疾病地圖絕對盛行率
1. 比較同時間各地區的盛行率 2. 有較高的對比度	1. 比較該年度之狀況 2. 利用整體顏色深淺觀察疾病趨勢

表 5-1：盛行率模式

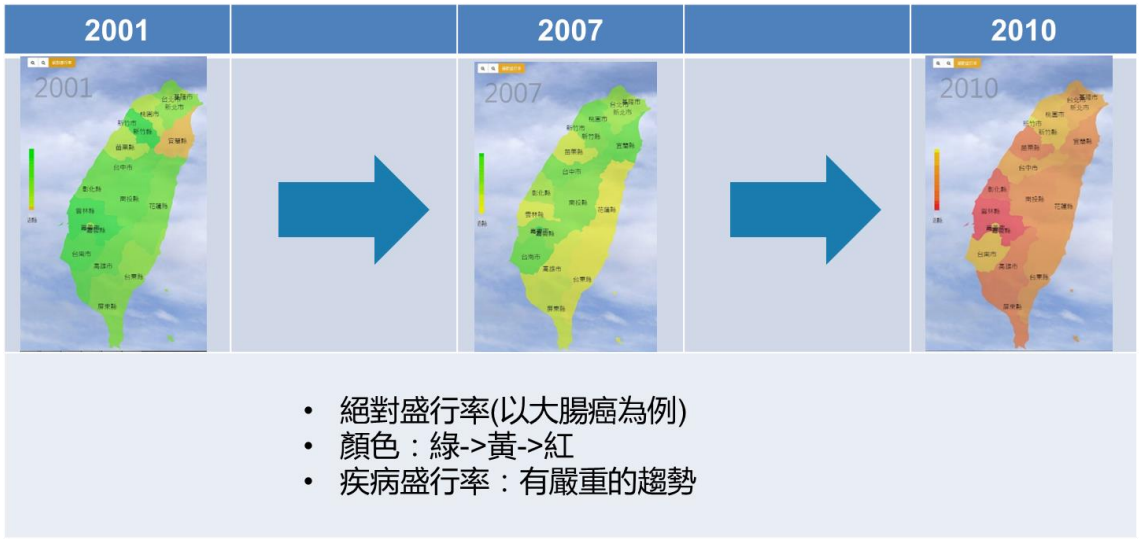


表 5-2：大腸癌絕對盛行率

doi:10.6342/NTU201601304

5-3-2 疾病趨勢圖

疾病趨勢圖負責呈現一個地區中不同時間的狀況及比較以補足疾病地圖中僅能呈現整體資料的不足，利用分頁將系統之疾病趨勢圖、疾病趨勢比較圖與疾病表格的功能分開，並且為了讓畫面更加簡潔，使用者一次只能觸發一個功能來使用；系統欲達到即時顯示的目的所以當使用者將滑鼠移入地圖中的縣市時，疾病趨勢圖就會動態更新資料，將該縣市之疾病通報數、盛行率以及各年齡層疾病分布之資料分別載入三組圖表如圖 5-5 所示，三張圖表設定為同步，讓使用者可以更清楚詳細地閱讀該地區的詳細資料，同時也利用小地圖呈現該縣市中鄉鎮的疾病分布狀況，如圖 5-8 所示；以大腸癌為例，在疾病地圖絕對盛行率模式中宜蘭區的顏色偏向黃色代表著盛行率是偏高的，將其盛行率與其他地區比較，並利用疾病趨勢比較功能如圖 5-6 所示，其原因可能是地下水遭砷的污染[17]有關，後來全面改用自來水變見其改善。

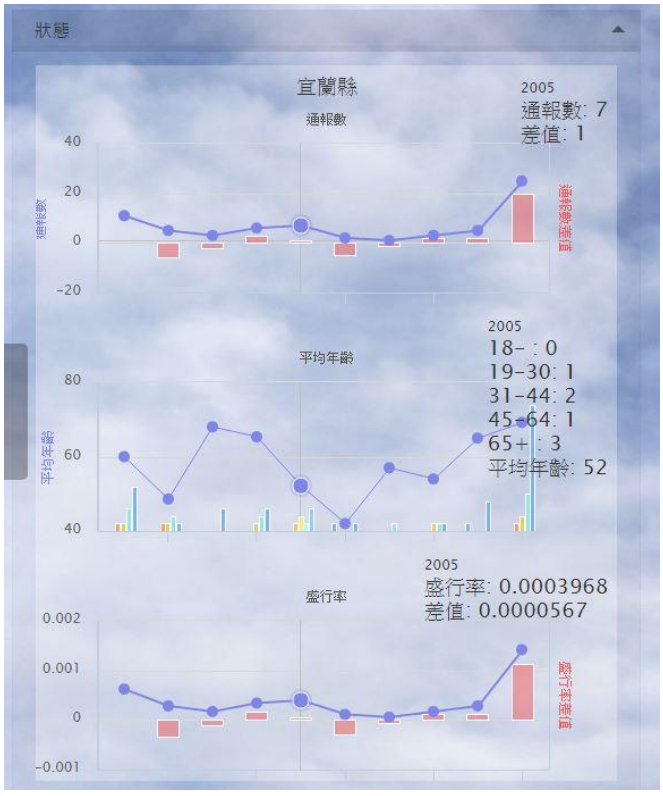


圖 5-5：疾病趨勢圖

doi:10.6342/NTU201601304

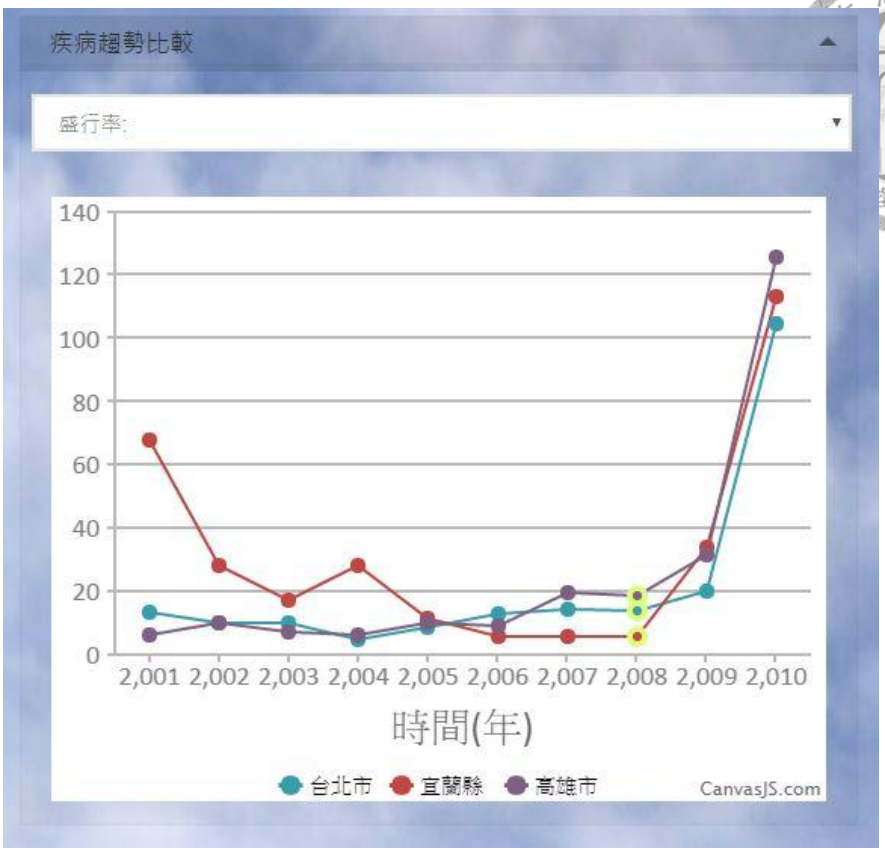


圖 5-6：疾病趨勢比較圖

圖 5-7：表格系統

doi:10.6342/NTU201601304

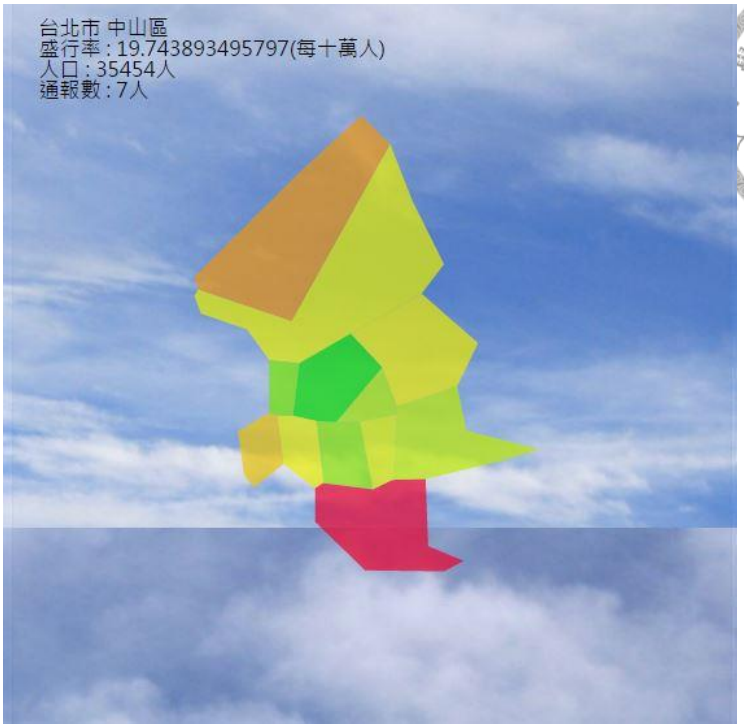


圖 5-8：縣市疾病分布圖

5-3-3 整合趨勢圖

傳統折線圖僅具有水平與垂直兩個象限，利用圓形物件的半徑可以增加第三個象限，系統將 x 軸設定為疾病資料相關可以選擇通報數與疾病盛行率，y 軸設定為水庫水質資料庫與自來水水質資料庫的屬性，圓形物件半徑設定為疾病通報數，使用者選取欲察看之地區，系統便會將該年度的疾病狀況依據年齡族群分類(18 歲以下、19 與 30 之間、31 與 45 之間、46 與 64 歲之間、65 歲以上)以五個圓形物件表示且綠色系與橘色系進行區別，可以使用自動展示功能或者透過拖曳時間軸來觀察物件移動的趨勢。

以大腸癌(ICD-9:154、1540、1541、1542、1543、1548)為例，觀察圓形物件高年齡層與低年齡層的圓形物件距離距離呈現加大的趨勢，臺東縣在年齡層 45 歲以上（45~64 與 65 歲以上）具有較高的盛行率，如圖 5-9 所示。將臺中市、臺北市與雲林縣作為城市與鄉鎮的實驗組與對照組，雖然從自來水質資料庫中硝酸鹽類 NO3N 之數值雲林縣之數值比起臺中市低很多，但雲林縣 45 歲以上（45~64

doi:10.6342/NTU201601304

與 65 歲以上）有明顯的疾病盛行趨勢，也許可以透過飲食習慣與年齡結構等其他因素來探討其致病原因。

圖 5-9：大腸癌-臺東

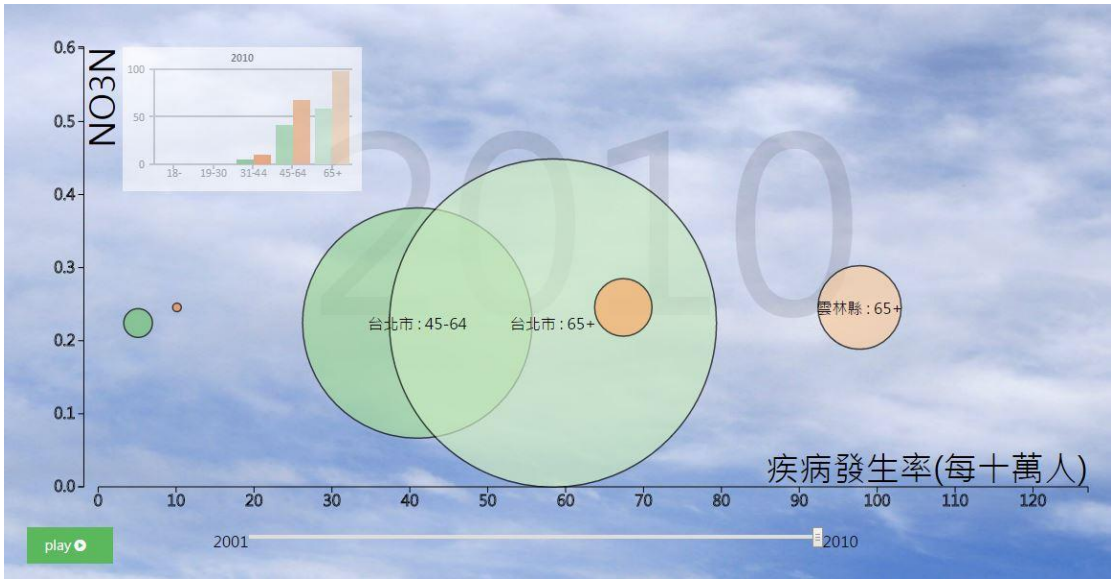


圖 5-10：大腸癌-臺北市、雲林縣

以呼吸道症候群（ICD-9:480～488）為例，將新竹縣與花蓮縣做比較，工業密集工廠林立的新竹縣無論 pm2.5、pm1.0 與 NOx 氣體[18]等具有汙染代表指標指

doi:10.6342/NTU201601304

數值均遠高於花蓮縣,或許新竹縣 18 歲以下之兒童具有相對花蓮縣較高的機會獲得呼吸道疾病。

5-4 疾病趨勢模組

希望能讓系統有更彈性的擴充空間，所以將整合趨勢圖模組化，其目的是讓水質資料庫或者空氣抽樣資料...等環境資料庫能用更適合的方式整合健保資料庫的疾病資料呈現給使用者，讓資料具有更好的研究價值，利用 MVC 架構容易擴充的優點，系統會根據查詢所使用的疾病代碼(ICD-9)所對應的 collection 名稱作為依據尋找對應的預測模組，若沒有找到則選用通用模組來取代，並在結果呈現的階段將預測模組或者功能模組嵌入於系統中。

doi:10.6342/NTU201601304

第六章　結論與展望

應用 NoSQL 資料庫 MongoDB 處理巨量健保資料，系統利用分散式技術查詢健保資料庫，解析回傳的查詢結果，將這些訊息利用疾病地圖以及趨勢圖…等視覺化工具呈現給使用者，欲讓使用者能夠探討疾病與環境數據潛在的關聯性，所以設計動態的方式載入環境資料庫(水庫水質資料庫、自來水水質資料)並在整合趨勢圖中呈現，讓使用者可以透過此工具探討疾病的趨勢以及隱含意義。

6-1 結論

本研究以視覺化為主軸，利用 MVC 架構建立此系統，並將功能模組化讓系統可以有彈性地選擇適當的模組以達到最好的呈現效果，將全民健康保險研究資料庫 1996 年至 2010 年 15 年間的百萬人抽樣承保資料,數量相當可觀可以以巨量資料的規模來處理，利用 NoSQL 資料庫 MongoDB 庫建立分散式資料庫系統，無綱要的特色讓每筆文檔(document)擁有很彈性的結構，讓每位承保人可以擁有不同數量的就醫紀錄，也因此在查詢時可以將就醫資料一次取得。

以使用者輸入的疾病代碼作為查詢條件，利用分散式技術 MapReduce，透過 map 函式將查詢分散為更小的程序以達到分散的效果，利用 reduce 函式將原始的歸人檔以地區代碼(reg_zip_code)歸檔，並僅將符合條件的承保人資料回傳給視覺化平台，建立 cache 系統以減少資料庫伺服器的存取，並顯著地增加系統效能。

視覺化的核心為用最精簡的畫面傳達最精準的資訊，同時為了兼顧完整性系統用動態的方式呈現疾病在時間與空間上不同面向的資訊，將全民健康保險研究資料庫的疾病資料包含:疾病的盛行率、各年齡族群之人數與平均年齡以及通報數等各項數值視覺化，透過疾病趨勢圖觀察時間面向地區的疾病詳細資料，利用控制器自動展示的功能讓時間演進，透過疾病地圖呈現空間資料，在相對盛行率與

doi:10.6342/NTU201601304

絕對盛行率模式中切換讓使用者可以瞭解該疾病在臺灣整體趨勢或是某年度的分布狀況，透過顏色變換觀察盛行狀況的改變來判讀疾病的趨勢，透過疾病趨勢圖的三個模組可以觀察縣市的詳細疾病資訊、鄉鎮分布狀況並且能比較不同區域的疾病資訊；利用非同步的方式讓使用者載入其他資料庫，例如:水庫水質資料庫、自來水水質資料庫…等，讓使用者選取想要觀察的屬性套入整合趨勢圖，為了提高資料的可讀性，利用事件例如:點擊、滑鼠移入甚至是拖曳…等讓使用者與系統互動。

透過這套疾病地視覺畫圖系統可以觀察疾病的分布現象，但由於年齡分層皆採用世界衛生組織年齡劃分依據，特定疾病可能需要做適當地年齡調整以達到最適當的顯示效果，若需要進一步的探討與研究分析則可以利用統計或者資料探勘等其他方法達成。

doi:10.6342/NTU201601304

6-2 展望

健保資料庫的承保人資料均為個人長期的追蹤病例，但沒有地區人口資料、生命統計資料以及不包含民眾自費的醫療項目，在使用上功能仍然不夠完善，若能將資料擴充或者與個人就醫紀錄串接起來，將可以有更多元的系統用途讓檔案更具價值。

目前查詢所使用的疾病代碼(ICD-9)較適用於從事醫療工作的使用者,未來希望系統能用更多的角度查詢健保資料庫，例如:用藥，讓從事藥物相關研究的使用者也可以使用此系統;整合更多環境資料庫,例如:空氣品質監測資料、懸浮微粒、土壤抽樣資料，利用系統的可擴充性建立更多預測模組以及功能模組，透過這套系統觀察疾病與不同環境數據之間的關聯性，期望能輔助解決社會上環境相關的問題。

此外更需要提升使用者的使用者體驗，除了設法提升系統效能以減少系統查詢的時間外,建立更簡潔的使用者介面讓使用者透過更簡單的操作傳達更多訊息。

doi:10.6342/NTU201601304

參考文獻

1. Tai, Y.-M. and H.-W. Chiu, *Comorbidity study of ADHD Applying association rule mining (ARM) to National Health Insurance Database of Taiwan.* international journal of medical informatics, 2009.
2. Lay, J.-G., K.-H. Yap, and W.-J. Chen, *地理資訊系統應用於登革熱疫情防治之檢討與建議.* Environment and Worlds, 2005.
3. *衛生福利部疾病管制署, 衛生福利部疾病管制署傳染病統計資料查詢系統.* 2015.
4. Ken Ka-Yin Leea, W.-C.T., Kup-Sze Choi, *Alternatives to relational database: Comparison of NoSQL and XML approaches for clinical data storage.* ELSEVIER, 2012.
5. *mongoDB.*
6. *ESRI Shapefile Technical.* 1998.
7. Chang, C.-L., *The research and development of disease mapping in Taiwan.* 2006.
8. *健保資料庫*. Available from: http://nhird.nhri.org.tw/.
9. 行政院環境保護署環境資源資料開放平台.
10. 行政院環境保護署環境資源資料庫. Available from: http://erdb.epa.gov.tw/.
11. Michael Bostock, V.O., and Jeffrey Heer, *D3: Data-Driven Documents.* 2011.
12. William A Mattingly, R.R.K., Julia H Chariker, Tim Weimken, Julio Ramirez, *An iterative workflow for creating biomedical visualizations using Inkscape and D3.js.* BMC Bioinformatics, 2015.
13. *Scalable Vector Graphics (SVG) 1.1 (Second Edition).* W3C, 2006.
14. *鄉鎮市區界線(TWD97 經緯度)*. Available from: http://data.gov.tw/node/7441.
15. Butler, H., *The GeoJSON Format* 2015.
16. dataUsman Iqbal, C.-K.H., Phung Anh (Alex) Nguyen,Daniel Livius Clinciu, Richard Lu, Shabbir Syed-Abdul,Hsuan-Chia Yang, Yao-Chin Wanga, Chu-Ya Huanga,Chih-Wei Huang, Yo-Cheng Changa, Min-Huei Hsu,Wen-Shan Jian, Yu-Chuan (Jack) Li, *Cancer-disease associations A visualization and animation through medical big data.* ELSEVIER, 2015.
17. Jin-Jing Lee , C.-S.J., Sheng-Wei Wang , Chen-Wuing Liu, *Evaluation of potential health risk of arsenic-affected groundwater using indicator kriging and dose response model.* 2007.

doi:10.6342/NTU201601304

18. Miao-Ching Chi, Y.-L.H., Yu-Chun Wang, *The Effect of Ambient Air Quality on Respiratory Diseases in Taiwan.* 2010.

doi:10.6342/NTU201601304