

# 雲端運算基礎架構下之地理資訊系統服務元件架構

## Service Component Architecture for Geographic Information System in Cloud Computing Infrastructure

王君如<sup>1</sup>、林峰正<sup>2</sup>、袁嵐焜<sup>3</sup>、辜文元<sup>4</sup>、周天穎<sup>5</sup>

<sup>1</sup>逢甲大學都市計畫與空間資訊學系研究生, <sup>2-5</sup>逢甲大學地理資訊系統研究中心

<sup>1</sup>st\_ruby@gis.tw; <sup>2</sup>francis@gis.tw; <sup>3</sup>peter@gis.tw; <sup>4</sup>cool@gis.tw; <sup>5</sup>jimmy@gis.tw

### 摘要

架構於雲端運算基礎上,本論文提出一個基於服務元件架構(SCA)的地理資訊系統模型。地理資訊系統整合不同類型的外部資料,帶來更廣泛、全面的視野,提供管理者在緊急情況下來做決策。過去我們已經成功的整合 SCA 於自動販賣機的服務研究,本論文則將此架構修改並整合於 GIS 的相關服務,除了大大地降低了程式的重複開發,更重要的是提供用戶一個全新的體驗,同時也解決了地理資訊系統的異質服務元件整合性議題,如:環境監測、遙感探測影像的處理、地圖服務、區域基礎的服務等等。而對於處理大規模資料的運算服務,我們則利用雲端的分散平行運算概念來處理,實作了一個呼叫水利署雨量站資料的 Web 服務,採用 IDW 內插法來求得未知雨量站雨量資訊,並成功在叢集環境下,運用 MapReduce 模型計算三種不同解析度的網格資料,提供台灣雨量計算資訊服務。研究顯示雲端分散式處理大範圍的網格運算,較單機最高減少為 1/3 的運算時間量,最後,在雲端運算架構下,導入服務元件架構於 GIS 的成功,可提高元件模組開發的效率,減少不必要的工作和時間,使地理資訊系統更加穩定,具有可擴充運算處理能力。

關鍵詞: 服務元件架構(SCA)、地理資訊系統(GIS)、雲端運算、分散式程式設計模型(MapReduce)、距離平方反比法(IDW)。

### Abstract

This paper proposes a service integration model of service component architecture (SCA) applied in services of Geographic Information Systems (GIS) based in cloud computing infrastructure. GIS integrates the different types of data to bring a broader, more comprehensive view to decision makers in the emergency. We have successfully integrated SCA into the domain of vending machines in the past. Readily, we focus on using this architecture to integrate related services of GIS, substantially reduce the duplication of development, and provide users with a completely new experiential usage. Meanwhile, we also address many components that can be integrated with system scaling up in GIS, such as environmental monitoring, remote-sensing

image's processing, Map services, location-based services, and so on. We deal with massive data by computing service though distributed computing in the cloud and call for one web service, which can obtain rainfall information from Water Resources Agency in Taiwan (WRA). This paper also employed interpolation computation by IDW to estimate values of unobserved rainfall information and succeed to compute the raster data of three different resolutions by MapReduce model and provide information service counting rainfall of Taiwan on clusters. According to this research, processing massive raster data by distributed computing in the clusters can reduce one-third times than a single machine. Finally, to import SCA for GIS can increase efficiency of component modularize development, reduce unnecessary work and time consuming, and make proposed system more stable and scalable.

Keywords- Service Component Architecture, Geographic Information System, cloud computing; MapReduce, IDW.

### 第一章、前言

地理資訊系統(GIS),是一套方便製作圖資及有效執行空間分析的電腦工具[1],它提供更有效率且系統化的方法來處理相關元素,讓你視覺化的理解地理空間資料。而 GIS 系統設計師,設計了許多有用且重要的 GIS 應用程式,提供終端用戶操作與決策使用,例如:土地利用和土地覆蓋的資源管理、適地性服務、災害管理,甚至都市及區域計畫。若要了解更多 GIS 的介紹,請參考 GIS 百科全書中的相關技術與案例,其著重於各類地理資訊系統的計算概念 [2]。

地理資訊系統概念無所不在,可以嵌入到既有的系統上,或透過外部呼叫 web service 方式,使用第三方 GIS 服務。而要滿足系統高效能與時程限制,最重要的想法是只要存取你所需要的特定功能即可,而不是存取一個完整的地理資訊系統 [3]。為了做到這點,我們將軟體元件從完整系統中獨立抽離,就像組合積木般,可以組合出不同的應用[9]、[10]。非常幸運地,服務元件架構(SCA) 框架,可以將元件組裝到應用程式中,且它的標準提供了具有彈性及可擴充的方法[4]、[5]。此外,SCA 通常由系統設計師以邏輯模組塊組合企業功能裝置的元件,並且在雲端環境下提供一個簡單的程式

設計模組來設計、組合、管理分散式應用程式建構。

SCA 有四個重要的概念[4]：服務、元件、組合、應用領域。在分散式的網域上支援高層次的功能，例如容錯性和可靠度特性，平行運算可以分散到不同的實體機器上，也可能是在虛擬機器上 (VMs)。

雲端運算運用虛擬化技術[6]，其動態資源配置能力及分散計算模型，讓你可以快速的建立虛擬機器且有效率的管理儲存與大量運算。因此，組合 SCA 和雲端運算將產生協同效應，使你的應用程式更靈活和強大。也就是說，應用程式建立在雲端上，在同一時間上可以使用不同的元件，更進階充份利用不同元件來產生新的組合元件。

我們已經成功的整合 SCA 在自動販賣機的領域[7]，以及提出了一個框架在雲端處理大量的遙感探測圖[8]。因此，我們著重使用這個架構去提供多種 GIS 的服務組合可能性研究，除了大大的減少元件重複開發，達到系統快速部署目標，期許在 GIS 領域上，提供一個全新的使用體驗。

本論文後續的內容如下，第二章透過相關的作業介紹 SCA 的概念，第三章探討以 SCA 為基礎的地理資訊系統應用程式，接著在第四章介紹 14 個元件和相關的複合服務，在第五章，我們提出並實作以 SCA 為基礎的真實案例—台灣雨量分佈資訊，而第六章比較單機與雲端叢集的運算效果，最後第七章為結論。

## 第二章、SCA 介紹

近幾年來，服務導向架構提供互通性服務系統架構，但如何實踐它便顯得有些困難，而 SCA 是一個以 SOA 為標準的技術。

SCA 所定義元件和組合是一個很好的技術，而且它可以使您的應用程式更容易編寫和維護，並讓管理者可直接更改應用程式的結構，不需更改應用程式本身。目的是支援更多元的技術以實現服務元件，並提供開發人員一個使用服務的簡單程式設計模型。這個不僅包含了特定的程式設計語言，而且還有相關框架，及其他基於這些程式語言的擴充系統。

在 SCA 說明書(2009) 和 SCA 白皮書(2006)

中，定義 SCA 是一套建構於 SOA 的應用規範之上，介紹如何建立元件，與如何組合那些元件到完整的應用程式，當中提供了一個建置應用程式模型，也解決了 SOA 的問題。SCA 允許開發人員著重在撰寫企業邏輯，然而，SCA 遵從現有的標準，以保存既有的中介軟體和工具，這個方法是一個以現有的方案為案例。

SCA 所提供的基礎結構積木是參考服務的元件，此外，SCA 支援服務元件的建立和重複使用，被設計為獨立的語言，以 Java、C++和 BPEL 來執行。

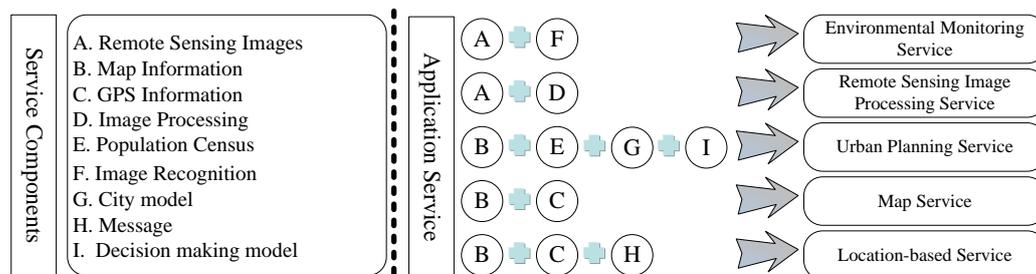
模型也是關於遠端元件之間的通訊協定的使用，同樣地，服務可以使用不同的語言來描述，SCA 可以組合元件以獨特的技術建置而成的。此外，使用 SCA 定義的框架下，元件的性能和互動裝置可以被配置在不同的情況。

Apache Tuscany [18]是一個 Apache 軟體基金會開發的原始計畫，由 Apache2.0 認證的網站 (<http://tuscany.apache.org>) 所免費提供的。實作細節可以參考 SCA 的教學課程[16]、[17]，在雲端上也可以執行 Tuscany[5]，並藉由呼叫 Tuscany APIs 來管理結構、建置、以及每個節點的生命週期。

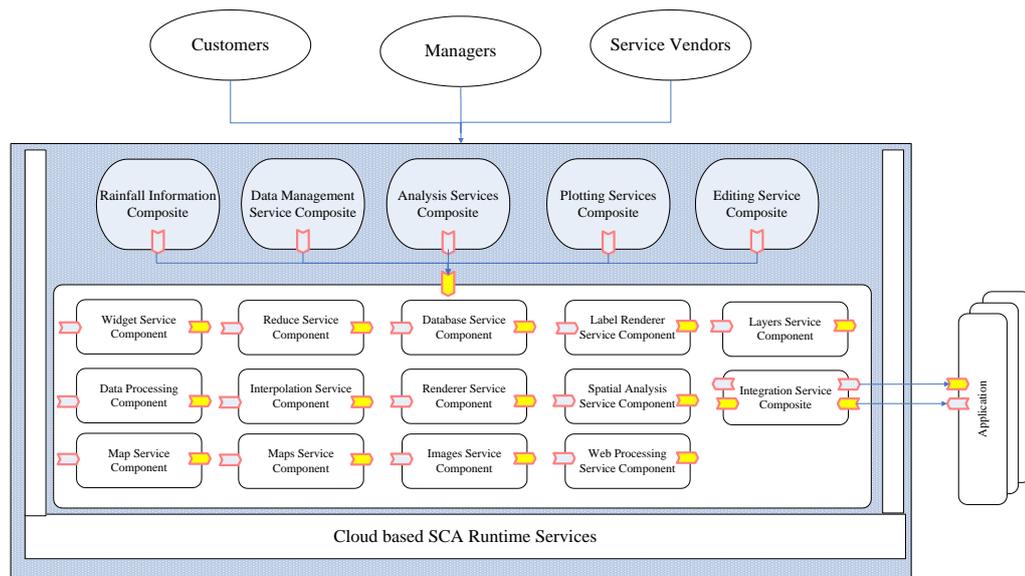
## 第三章、應用 SCA 於地理資訊系統的應用

目前有許多 GIS 專業人士使用的地理資訊系統，例如：都市計畫者使用 GIS 來發展和監督都市的規劃、市場分析應用 GIS 來預測和規劃企業的成長、科學家利用遙測影像來輔助土地的覆蓋和辨識[13]。在軟體的重複使用的概念下，元件和開發者可以由各種不同的來源組裝元件給 GIS 工具箱，以滿足不同的需求[3]。

每個元件通常實現企業程式邏輯，公開為一個或多個服務。一個服務提供多個作業方式，可以讓該組件的客戶端存取它。而如何描述服務取決於實現元件的技術，介面和查詢可以讓一個元件和其他服務溝通。服務元件架構提供功能來表示企業的邏輯想法，做為一個可以重複使用的元件，當要被組合一個應用程式或完整的解決方案時，可以很容易的被達成，圖一為架構 SCA 概念的 GIS 組裝服務的 5 個案例。



圖一 GIS 應用於組裝服務的例子



圖二 基於 SCA 於地理資訊系統和相關的服務架構

#### 第四章 地理資訊系統的服務元件架構

我們擴充以 SCA 為基礎的服務概念成一個大型的地理資訊系統和相關的服務，描述如上圖二，有 14 個元件來支持 5 個組合的元件，其中包含了雨量資訊組合、資料管理組合、分析服務組合、繪圖服務組合以及編輯的服務組合。這些組合元件可以藉由管理自己的各種服務元件來實現整個服務。在本文中，我們實現了以 SCA 為基礎的案例(雨量資訊組合)將在第五章之後提出。在此之前，底下將介紹每個元件：

##### 一、介面服務元件

介面服務可以讓終端使用者在使用者介面互相交換圖資，這樣他們可以輸入、編輯原始資料、存檔及分析儲存的資料，並呈現和顯示地理空間資料。隨著時間的改變，高階的介面可以由控制器搭配動畫顯示地圖。因此，這個元件是終端的使用者和 GIS 服務的溝通橋梁。

##### 二、資料處理元件

這個元件可以用於預先處理網格資料或向量資料，並可呼叫其他元件，例如 Map 和 Reduce 服務元件，來獲得進一步結果。

##### 三、Map 服務元件

這個服務元件可以讓使用者定義 MapReduce 程式語言模型，並帶有一個鍵值作為輸入，並產生一組鍵值作為輸出。在這裡，Map Service 的轉換由輸入格式提供的鍵值，那是輸出格式所需要的 [11]。因此，這個元件指定了兩個功能：有 Map 函數和 Reduce 函數。

##### 四、Reduce 服務元件

Reduce 函數可輸出 0 或 1 的值給任何一個鍵，中間值提供給 Reduce 函式通過一個迭代器

[11]。同樣地，Reduce 服務在這個框架下可能因為不同的 GIS 應用程式而改變。

##### 五、內插的服務元件

GIS 的專家經常利用內插法來估計未知的地區，例如：區域性的降雨或受污染的區域。這個元件提供了內插服務和特別的計算，例如：數學公式(距離反比加權法，IDW)、地理統計應用技術(克利金)和其他 [20]。

##### 六、Maps 服務元件

Maps 服務提供地圖顯示功能，你可以以一個標準協議發布地圖。應用最廣泛和受歡迎的是 OGC 的網路地圖服務(WMS)，其概述了通訊的方法，讓不相容的軟體要求且提供預先安裝的圖影像(編輯包含向量和網格資料的地圖影像)給提出需求的客戶端。

##### 七、資料庫服務元件

企業可開發先進的技術來儲存和管理資料庫，地理資料庫可以利用資料庫管理系統(DBMS)，將系統文件完整地存儲，此元件正是幫助你透過相關的 web service，來傳送資料庫服務。

##### 八、輸出服務元件

輸出服務元件通常依資料屬性來創造不同類型的圖表。這個元件還允許以管理者的身分，在網路上管理圖表並且發佈於網頁上。

##### 九、影像服務元件

這個元件允許個人或系統上傳遙測影像，並執行影像處理，像是分類或分群、或者影像鑲嵌。

##### 十、標籤輸出服務元件

如果你有一個都市的地圖，並且想要在地圖上

貼上名稱標籤來表示為著名的旅遊景點。標籤輸出服務元件通常和 Maps 服務元件一起執行，讓地圖圖徵都標有屬性資料。

#### 十一、空間分析服務元件

並非所有的地理空間問題都可以被劃分成較小的元件問題，因此這個元件有助於在單機上或分散式環境分析地理資料。

#### 十二、Web 處理服務元件

OGC 在 2007 年提出的網路處理服務(WPS)介面規範標準，並且描述了一個標準化的方法，來發布和執行以網路為基礎的程序提供給任何的地理資訊處理使用。在這個元件，您的介面應該有三種操作能力：接收能力、描述過程、執行。

#### 十三、Web 處理服務元件

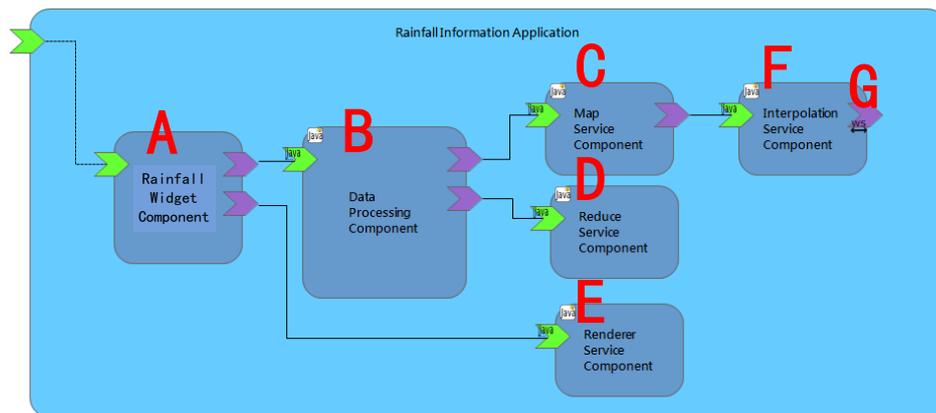
圖層服務元件可以幫助使用者，加入額外的資訊到圖層上。因此，您可以使用這個元件來增加地圖上的特定圖層(影像圖層或追蹤圖層)。

#### 十四、整合服務元件

在一些情況下，內部以計畫為基礎的應用需要與外部的應用程式進行通訊，設計師可以建立整合服務和其他網路服務、JMS、CORBA、RMI、或 REST [16]應用程式做關聯。

### 第五章、SCA 實際案例

每年夏天有許多颱風襲擊台灣，並帶來很大的雨量，為了要計算累積降雨量，降雨資料便是一個 GIS 很重要的應用，是一個很好的資料來呈現組合的概念，這個系統在颱風季節可以保護居民免受水患，也可以幫助政府在過程中做出重大的決策。因此，我們提出並實作是基於 SCA 相關的降雨資訊應用。本文將不討論關於細節的部分，但我們列了 7 個部分，分別於下圖三的 A 到 G，其利用 Eclipse 表示 SCA 的複合圖[19]，安裝了 SCA 工具和 Tuscany。實驗環境為 3 台實體機器，硬體配備 (CPU: i5-2400、RAM: 12GB)，作業系統為 Ubuntu Linux 11.10 及搭配 Hadoop ver. 0.20.205。



圖三 以 SCA 為基礎的降雨資訊應用

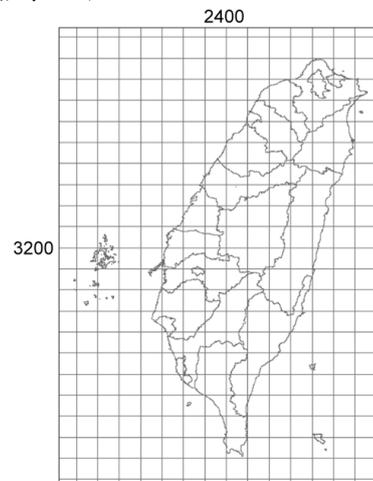
#### A、降雨量介面元件

在這個組合架構中，一個 web 的視窗介面被安裝並且在網頁執行。Tuscany 執行時支援整合 HTML 和 Web2.0 用戶端的應用程式，例如：Ajax、Flex 和整合 scripting 語言，例如：Ruby、Groovy、and JavaScript [16]、[17]。這個視窗介面將會呼叫資料處理元件，來處理網格資料和輸出服務元件，以呈現降雨空間分佈的結果。

#### B、資料處理元件

此元件用來處理網格資料，如圖四所示，一個網格資料被劃分成寬度 2400 個，高度 3200 個的網格。每個值都代表了一個 125x125 公尺的網格單元降雨點。此外，MapReduce 是一個軟體框架叢集，可以根據不同的應用做修改，來支援分散式計算資料群。因此，這個元件是負責處理網格資料(傳送網格資料到一個文字檔案)、分配空間資料到 Map 服務元件、藉由 Reduce 服務元件收集已經處理完的資料。Hadoop 的 MapReduce 是一個以文字方式

處理，所以基本上運算的結果必須被轉換到另一種格式，藉由輸出服務元件轉為人類可讀的格式，例如 JPG 影像文件。



圖四 台灣地圖切割，每一網格為 125x125 公尺

### C、Map 服務元件

這個元件通常被用來讀每個單元的網格位置到地圖函數，並且呼叫內插服務元件以 IDW 公式來取得每個單元值所預估的雨量值。圖五是 Map 服務的虛擬程式碼。

```

//*****
// Map Service Component
//*****
map(string key, string value)
// key: the Grid key that will be evaluated
// value: the Grid coordinates that will be evaluated
begin
// Compute each evaluated point in grid data sets, EvaluateP
Refer to Interpolation Service Component to get EvaluateP.
Emit ( key, EvaluateP)
end
    
```

圖五 Map 服務的虛擬程式碼

### D、Reduce 服務元件

Reduce 服務元件是負責整合每個網格單元的 IDW 結果，並由輸出服務元件產生一個輸出文件，圖六為虛擬程式碼。

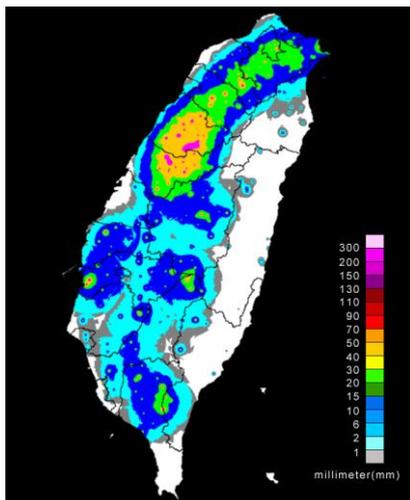
```

//*****
// Reduce function
//*****
reduce(string key, Iterator value)
// key: cell key
// value: The result of IDW
begin
Emit(AsString(value))
end
    
```

圖六 Reduce 服務的虛擬程式碼

### E、降雨量畫面呈現服務元件

為了直接觀察降雨量資料，輸出服務元件通常被用來顯示網格資料。圖七呈現 IDW 服務估計台灣空間雨量分佈。



圖七 利用 IDW 估計降雨的空間分佈情況

### F、內插的服務元件

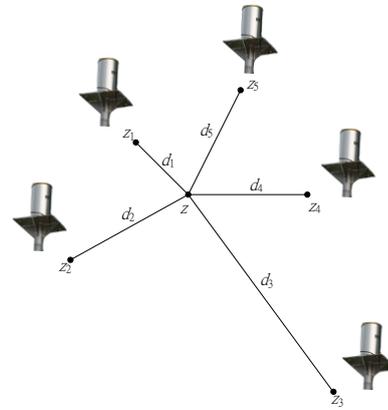
空間內插的演算法有很多種，例如：反距加權法(IDW)、splines、克利金法、趨勢分析和傅立葉級數法。

本研究使用距離反比加權法(IDW)[14]，[12]

來推估每個網格降雨量。距離反比加權法(IDW)[14]是假設每個測量點有一個局部影響值，會隨著距離而消滅，因此，在附近的點給較高的權重，而較遠的距離則給予較小的權重。此外，Revesz 和 Li 使用距離反比加權法(IDW)將空間內插視覺化[15]。IDW 的內插公式通常用在內插服務元件如下，圖九為 IDW 公式示意圖：

$$z_j = w_j \sum_{i=1}^N \frac{z_i}{d_{ij}^\alpha}, \text{ where } w_j = \frac{1}{\sum_{i=1}^N 1/d_{ij}^\alpha} \quad (1)$$

其中  $Z_j$  是座標  $(x_j, y_j)$  網格的預測值， $Z_i$  是雨量站的座標  $(x_i, y_i)$  已知值， $N$  是  $Z_j$  附近 (25 公里內) 數個已知點，圖八為其關係示意圖。 $w_i$  是權重分配給每個已知點給  $z_j$ ， $d_{ij}$  是兩點間  $(X_i, Y_i)$  和  $(X_j, Y_j)$  的歐氏距離 (Euclidean distances)， $\alpha$  等於 2，其影響  $w_i$  在  $z_j$  的加權，圖九為內插服務元件的虛擬程式碼。



圖八 降雨內插公式示意圖

```

//*****
// Interpolation Service Component by IDW
//*****
Input the Evaluate Point, EvaluateP
Find the nearest rainfall stations (within 25 km), NearestSet
Calculate IDW Weight, w

for each p in NearestSet
begin
w += 1/ distance(p, EvaluateP)
end
// Evaluate cell value by IDW
for each p in NearestSet
begin
EvaluateValue += w * (1/distance(p, EvaluateP)) * p.value
end
    
```

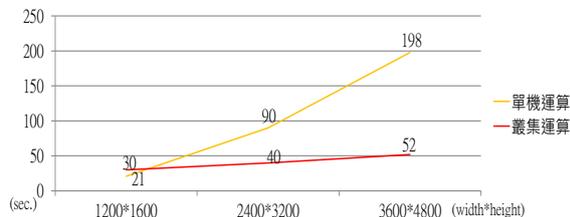
圖九 內插的服務程式碼

### G、呼叫第三方 Web 服務

雨量資料是由台灣的中央氣象局來蒐集，其中包括了 500 個雨量站的資料，然而，這些資料不能直接提供給民眾，因此我們必須透過經濟部水利署得到雨量資料，所以內插服務將藉由 SCA 的 binding.ws 與水利署做通訊。內插服務元件將連線到 Web 服務以取得雨量站資料。

## 第六章、雲端叢集運算結果

Hadoop 執行分散運算時，由於 Job 初始化時間較長，對於少量運算(1200\*1600)，分散運算不見得佔有優勢，但隨運算量增加，處理較大的網格資料時 (2400\*3200-3600\*4800)，叢集運算就有明顯的優勢，由圖十可知叢集運算較單機運算時間上，最高可減少為 1/3 運算時間量。



圖十 單機與叢集運算比較

## 第七章、結論與未來方向

本研究首先提出了以雲端為基礎的 SCA 整合模型，解決 GIS 異質元件整合議題，而分散式的 SCA 配置在 Apache Tuscany 的平台上，支援容錯性和可靠度，可以使用不同機器所開發的元件，因此我們探討了基於 SCA 的 GIS 相關元件，其中複合式應用包括了降雨量介面、資料處理、Map 服務、Reduce 服務、降雨量畫面呈現服務、內插服務、呼叫第三方 Web 服務，這種複合式應用，幫助我們減少服務的超載和提高系統的效率，未來在此架構下，將會產生更多 GIS 的應用。最後實作的降雨資訊內插運算，我們比較單機與叢集的運算效能，證實雲端的可行性與高效能。

### 致謝

本研究由國科會 NSC100-2625-M-035-003 和 NSC100-2119-M-035-001 補助。

### 參考文獻

- [1] W. Qihao, Remote Sensing and GIS Integration, Theories, Methods, and Applications, McGraw-Hill Professional, 2010.
- [2] S. Shekar and H. Xiong (Eds), Encyclopedia of GIS, Springer, 2008.
- [3] R. Hartman, Focus on GIS Component Software: Featuring ESRI's MapObjects, OnWord Press, 1997.
- [4] J. Marino and M. Rowley, Understanding SCA (Service Component Architecture, Addison-Wesley Professional, 2009.
- [5] S. Laws, M. Combella, R. Feng, H. Mahbod, and S. Nash, Tuscany SCA in Action, Manning Publications Co., 2011.
- [6] G. Toraldo, OpenNebula 3 Cloud Computing, Packt Publishing Ltd., 2012.
- [7] F. C. Lin, Y. S. Lee, C. H. Hsu, K. Y. Chen, and T. C. Weng, "Service Component Architecture for Vending Machine System in Cloud Computing Infrastructure," IEEE International Conference on e-Business Engineering (ICEBE '09), IEEE Press, Oct. 2009, pp. 591-595.
- [8] F. C. Lin, L. K. Chung, W. Y. Ku, L. R. Chu, and T. Y. Chou, "The Framework of Cloud Computing Platform for Massive Remote Sensing Images," unpublished.

- [9] C. Granell, L. Díaz, and M. Gould, "Service-oriented applications for environmental models: Reusable geospatial services," Environmental Modelling & Software, vol. 25, pp. 182-198, February 2010.
- [10] L. Braubach and A. Pokahr, "Addressing Challenges of Distributed Systems Using Active Components," Proc. of 5th International Symposium on Intelligent Distributed Computing (IDC - 2011), Netherlands, Oct. 2011.
- [11] T. White, Hadoop: The Definitive Guide, O'Reilly, 2009.
- [12] F. W. Chen and C. W. Liu, Estimation of the spatial rainfall distribution using inverse distance weighting (IDW) in the middle of Taiwan, Paddy Water Environ, DOI 10.1007/s10333-012-0319-1, 2012.
- [13] N. C. F. Codella, G. Hua, A. Natsev, J. R. Smith, "Towards Large Scale Land-cover Recognition of Satellite Images," Proc. IEEE Information, Communications and Signal Processing (ICICS 8th), IEEE Press, Dec. 13-16, 2011, pp.1-5.
- [14] D. Shepard, "A two-dimensional interpolation function for irregularly spaced data," 23rd National Conference ACM, 1968, pp. 517-524.
- [15] P. Revesz and L. Li, "Constraint-based visualization of spatiotemporal databases," Advances in Geometric Modeling, chapter 20, John Wiley, 2003, pp. 263-276.
- [16] Vincent Zurczak, "Towards Integrated SOA Development with Eclipse STP and Swordfish: SCA Tutorial," EclipseCON 2009, March 2009.
- [17] First steps with the SCA Composite Designer, available in [http://wiki.eclipse.org/STP/SCA\\_Component/SCA\\_First\\_Steps\\_With\\_Composite\\_Designer](http://wiki.eclipse.org/STP/SCA_Component/SCA_First_Steps_With_Composite_Designer).
- [18] Apache Tuscany, <http://tuscany.apache.org/>.
- [19] Eclipse STP website, <http://www.eclipse.org/stp/>.
- [20] C. Childs, Interpolating Surfaces in ArcGIS Spatial Analyst, available in <http://webapps.fundp.ac.be/geotop/SIG/interpolating.pdf>.