

空間雲端運算平台架構設計

The Design of Spatial Cloud Computing Platform Architecture

辜文元
逢甲大學 GIS 中心
Wen-Yuan Ku
Email: cool@gis.tw

周天穎
逢甲大學 GIS 中心
Tien-Yin Chou
Email: jimmy@gis.tw

袁嵐焜
逢甲大學 GIS 中心
Lan-Kun Chung
Email: peter@gis.tw

摘要

隨著網際網路技術進步與創新，帶動雲端運算產業蓬勃發展，也成為未來資訊軟體的發展趨勢。雲端運算的掘起，讓企業重新思考是否還需要自行營運資訊機房，或者將既有系統及資料移轉至雲端平台上運行。由於長期的空間資料儲存與分析需要耗費大量的儲存空間及計算資源，早期空間分析仰賴單機作業方式進行，對於需要大量分析運算的空間資料而言，耗費較多時間成本執行運算工作。如何在複雜的空間分析運算上，提供資料分析速度以縮短決策分析所需時間，此為目前所面臨問題。雲端運算技術的提出，恰好解決這個問題，藉由雲端運算 MapReduce 架構，將巨量的空間資料切割成較小的資料單元，平行化進行資料處理與運算，來加速資料的處理與分析速度。

本研究以開放式雲端軟體 Hadoop 為基礎，提出一種具備巨量級資料分析能力的空間雲端運算平台架構，設計上遵循開放式大地空間協會(Open Geospatial Consortium, OGC)所提出的網路處理服務(Web Process Service, WPS)標準作為用戶端與空間雲端運算架構橋接介面，而為驗證架構之可行性，本研究以大範圍內插運算為例，驗證空間雲端運算架構概念可行性。

一、前言

自從 IBM 在 1960 年代推出第二代電腦後，地理資訊系統(Geographic Information Systems, GIS)的技術也隨之萌芽。1963 年世界第一套 GIS 軟體問世，主要應用於土地的利用和自然資源分布的調查上。GIS 經過多年的發展，GIS 已經是集電腦繪圖、遙感探測、電腦輔助設計和資料庫管理等 4 門領域的技術。GIS 資料經年累月長期累積下其資料量非常可觀，除了資料儲存面臨問題外，如何將這些巨量 GIS 資料進行有效率處理與分析更是一個大挑戰。

而從另一個分享的層面思考，傳統 GIS 軟體與資料通常被存放於單一個人電腦或伺服器上，容易衍生資料重覆建置及資料版本差異問題，同時業務單位為了滿足業務上需求，也必需求採購軟硬體設備，造成相同的軟硬體設備必需被不斷建置，造成重覆資源浪費。

近年來由於服務導向架構興起，帶領著 GIS 服務與應用到另一個分享層次。由 OGC 著手所制定

的 GIS 標準，就是希望藉此將分散在各地的資源可以在共同標準下進行資料交換並達成資料流通目的，例如:Web Map Service(WMS)或 Web Feature Service(WFS)就是目前最被常用的 2 種 GIS 資料交換服務。除了遵循 OGC 標準建立資料分析服務或，藉由 WebGIS 技術架設網站分享地圖資料或分析服務給其他使用者，也是目前最被普遍的資源分享模式。

但目前不論是使用 WebGIS 技術或發布 OGC WMS/WFS 服務，其根本基礎仍然為單一伺服器單處理序的運作模式，一旦同時使用者上線數增加或者分析資料量較大，就很容易造成了伺服器極大的負荷。而雲端運算技術的誕生，正好為大量使用者需求及巨量資料處理問題的解決方案。

二、相關研究

由於 GIS 資料增長速度遠比 CPU 運算速度提升來得快，傳統經由單機處理或分析資料漸漸不符合實際需求，因而分散式平行處理 GIS 架構逐漸受到重視。開放式雲端軟體 Hadoop 的誕生，加速了 GIS 在分散式平行處理的研究。Qichang Chen 等人在 2008 年提出 MRGIS[1]就是利用 Hadoop MapReduce 技術所實現出的 GIS 運算架構，前端透過 Python 與 Hadoop 進行整合。

此外，Nathan Kerr 在 2009 年也提出了 GIS 資料平行處理架構[2]，並同時比較單機運算與 Hadoop GIS 及 Cluster GIS 在執行上的效率，顯示出 GIS 分散式處理可以有效縮短資料分析處理時間。而由這些文獻中可以發現 Hadoop 是目前在 GIS 分散式平行處理相當好的發展平台，除了本身免費之外，Hadoop 也提供相當完整的 Framework API，提供開發人員可以自行發展出自己的運算服務。

三、雲端運算技術

雲端運算是基於網際網路的新型態服務概念，不論終端是那一種的設備，都可以透過網際網路提供即需即取的資源。由於資源是在網際網路上，而在系統架構圖中，網際網路通常以雲形表示(圖 1)。因此被形象化類比為雲端運算，雲端同時也是對底層基礎設施的一種抽象概念。雲端運算的資源是動態、可彈性擴充而且虛擬化的，透過網際網路提供。用戶不需要了解雲端基礎設施的細節，也無需直接進行控制，只關注自己真正需要什麼樣

的資源以及如何透過網路來得到相應的服務。

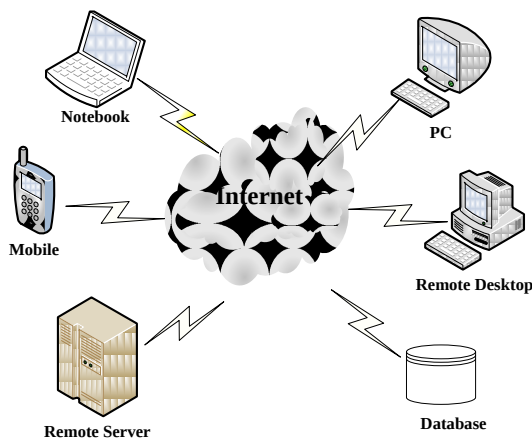


圖 1 雲端運算架構示意圖

在雲端運算中軟體和硬體都被抽象化視為資源型態，並且封裝成服務形式對外提供服務，使用者透過網路取得需要的資源，陳澄[3]依雲端服務類型，將雲端運算分為三種類型，不同類型雲端服務有著不同的用途，分別對映到不同的應用階層(圖 2)，由使用者自己依據需要來決定採用哪一種的雲端服務。

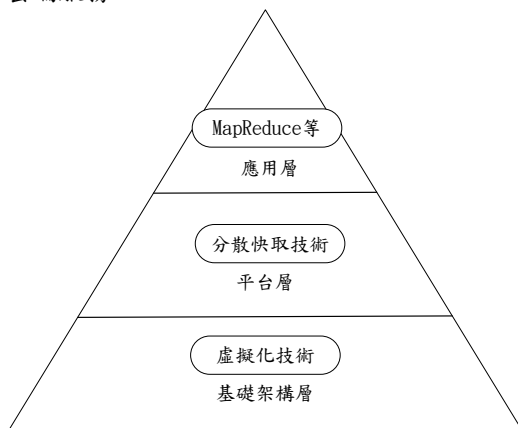


圖 2 雲端服務類型

1. 基礎架構雲：架構即服務(IaaS, Infrastructure as a Service)

將基礎架構透過虛擬化的平台整合成為一項服務提供給用戶，用戶不需要採購伺服器、網路設備及軟體，用戶可自己決定安裝的作業系統，透過虛擬化技術，用戶可以在資源不足時，隨時動態擴充需要的資源，當不需要資源時也可以隨時釋放，對用戶來說這種方式最彈性。例如：Amazon Elastic Compute Cloud (Amazon EC2) 就是 IaaS 最成功典型範例。EC2 透過 Xen 虛擬技術，提供了虛擬的主機執行環境及不同等級的運算環境，可讓用戶選擇 Linux、OpenSolaris、Windows Server 等作業系統及 Web 伺服器、資料庫軟體、開發環境。

2. 平台雲：平台即服務(PaaS, Platform as a Service)

IaaS 將硬體基礎當作服務，PaaS 則進一步的將

較為抽象的「平台」視為提供給用戶的服務，PaaS 提供執行平台，開發人員僅需遵守 PaaS 的開發規則與限制，例如：程式語言，資料儲存模式等，就可以將程式編譯後部署到雲端平台，開發人員毋需考慮程式執行效率及管理，PaaS 會負責監控應用服務的存取流量自動配置適當的資源，例如微軟的 Azure Services Platform 及 Google App Engine 都是 PaaS 的具體實現。

3. 應用雲：軟體即服務(SaaS, Software as a Service)

與 IaaS 與 PaaS 不同，SaaS 直接提供用戶所需要的應用服務，軟體不再是以往的軟體，而是以服務的方式存在，最大的好處是簡化了軟體的部署與維護工作，降低 MIS 部門的人力需求，未來幾年，以 SaaS 形式存在的軟體會越來越多，SaaS 也將是一股不能抵擋的潮流。舉例而言：Google Docs、Microsoft Exchange Online、Salesforce.com 等。

四、分散式平行處理技術-MapReduce

雲端運算資料處理技術首推 MapReduce[4]技術，MapReduce 是 Google 所提出的分散式平行運算模式，很適用在巨量資料的處理工作。MapReduce 的整個架構是由對映(Map)以及重組(Reduce)兩個步驟所組成，運作時系統會將資料進行切割，並分配給不同的運算主機進行處理，因此這些被切割出的資料就可以在不同主機上平行進行處理，而通常使用每一個分割的大小是 16~64MB。

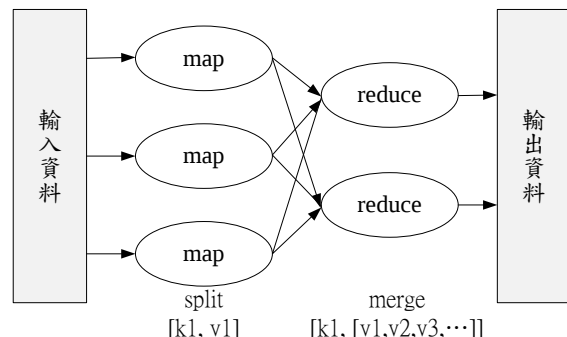


圖 3 MapReduce 運算架構示意圖

在執行 Map 過程中，系統讀入一組鍵/值(key/value)後，透過 Map 程序產生多組中間運算鍵值(intermediate key/value)，最後經由重組程序合併具有相同 key 的中間運算序列，產生最後的結果。

使用 MapReduce 有幾個特點：平行與分散式運算、具備容錯能力、執行狀態監控，這使得運算工作執行時可獲得高度可靠性。

五、空間雲端計算基礎架構

本研究提出的空間雲端計算基礎架構如圖 4 所示，架構上分為三個部份：雲端服務入口、雲端服務及雲端作業環境。

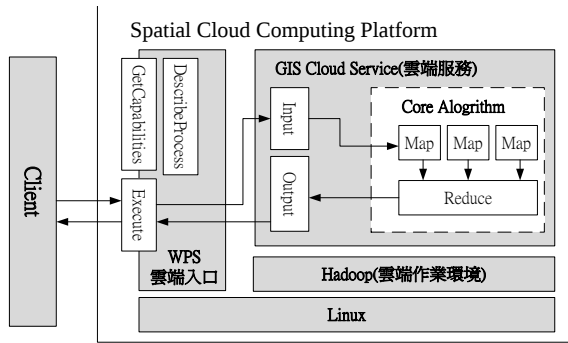


圖 4 空間雲端運算平台架構圖

1. 雲端服務入口

開放性架構是目前軟體架構設計相當重要的設計觀點，單純的將軟體視為元件，降低不同系統彼此間的耦合度，讓未來的系統整合及應用可以更具備彈性。因此在雲端服務入口導入了 OGC WPS 標準[5]，WPS 主要針對服務規範定義標準的介面，以便網路上各種異質應用系統可以進行整合、溝通及交換服務。WPS 介面如圖 5 所示，WPS 提供三個標準介面，GetCapabilities、DescribeProcess 及 Execute 三個介面。GetCapabilities 介面提供客戶端取得雲端服務基本資訊，DescribeProcess 介面則是提供雲端服務的說明，而 Execute 介面則提供了執行服務能力，趨動雲端服務的執行。

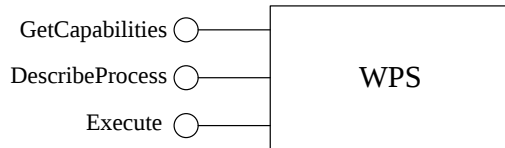


圖 5 WPS 介面圖

由於雲端服務計算時間會受到資料量與運算演算法複雜度的影響，因此架構設計上提供了同步與非同步的執行方式，在同步執行模式下，客戶端送出服務請求後，會等待雲端服務執行完成，並接收其運算後的結果；而非同步執行讓客戶端送出服務請求後，不用等待雲端服務執行完成，待雲端服務執行完成後，再通知客戶端來取用執行結果，適合用於需要長時間運算處理的服務。

而為了讓空間雲端計算基礎架構能夠具備彈性，在 Execute Request 訊息設計上，本研究制定訊息呼叫標準格式(表 1)。其中 DataInput 僅使用一個參數，命名為 Request，形態為字串；DataOutput 命名為 Response 形態為字串。DataInput 格式包含了幾個必要元素：ServiceID、Mode 及 Parameter，依據每一個 GIS 雲端服務需求彈性進行設計。

表 1 訊息呼叫標準格式

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<wps:Execute service="WPS" version="1.0.0"
xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsExecute_request.xsd">
<ows:Identifier>GISFCU.GISCloudPlatform.WPS</ows:Identifier>
```

```
<wps:DataInputs>
  <wps:Input>
    <ows:Identifier>Request</ows:Identifier>
  </wps:Input>
  <wps:Data>
    <wps:LiteralData>
      <WPSRequest>
        <ServiceID>服務識別碼</ServiceID>
        <Mode>Sync/Async</Mode>
        <Parameter>
          <使用者自行定義參數 />
        </Parameter>
      </WPSRequest>
    </wps:LiteralData>
  </wps>Data>
</wps:DataInputs>

<wps:ResponseForm>
  <wps:ResponseDocument storeExecuteResponse="false">
    <wps:Output asReference="false">
      <ows:Identifier>Response</ows:Identifier>
    </wps:Output>
  </wps:ResponseDocument>
</wps:ResponseForm>
</wps:Execute>
```

2. 雲端作業環境

於雲端環境建構上，使用由 Yahoo 支持的開源軟體 Hadoop 做為雲端作業環境，Hadoop 是以 java 為基礎發展出來的雲端軟體，實作了 Google 三項重要核心技術，分別為分散式檔案系統(HDFS)，分散式平行處理(MapReduce)及分散式資料庫(HBase)，目前 Hadoop 已有許多實際成功的商業應用案例，足以顯示 Hadoop 在巨量資料處理能力。

3. 雲端服務

雲端服務由開發人員依據業務特性及需求進行開發，由於雲端服務主要特性為解決大量資料處理問題，因此開發上必需針對資料格式進行規劃，再設計對映(Map)及重組(Reduce)演算法。當完成雲端服務開發並編譯完成後，上傳至空間雲端運算平台，就可以將服務掛載至空間雲端平台。

六、實驗測試

為驗證本研究提出之空間雲端運算基礎架構之可行性，本研究以空間內插運算應用服務為應用案例，將服務部署於雲端平台上進行測試，評估執行效能。本研究利用 3 部電腦建立實驗環境。電腦主機 AMD Phenom 2.3GHz 4 核中央處理器，記憶體為 4GB。主機作業系統為 ubuntu Linux 9.10 及雲端軟體採用 Hadoop 0.20.1 進行環境架構。

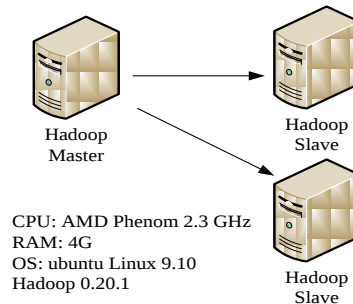


圖 6 空間雲端運算平台實驗配置圖

1. 內插服務實驗設計

本研究利用台灣地區既有的中央氣象局 400 餘座雨量站資料，推估台灣地區各地的降雨值，台灣地區南北長約 377 公里，東西寬約 142 公里，如果以網格解析度 100 公尺為例，則總共計有 535,340 個網格值必需要被評估計算，此時利用雲端運算的特性，透過對映程序將所要計算的網格分散到不同的運算主機上做處理，藉由分散式平行處理可加速網格計算處理時間，最後將經由重組程序產生分析後的台灣地區降雨量內插結果。

2. 資料格式設計

由於 MapReduce 運算時，資料必需經由 Map 程序將資料切割後方能進行分散式處理，MapReduce 讀取資料採用文字模式，亦即逐行讀取資料進行處理。因此配合 MapReduce 資料分割及讀取程序，當系統接收到處理工作時，會依據要分析範圍及網格解析度，預先產生網格資料檔，提供給分割程序運算使用。其資料格式如表 2 所示，坐標採用 WKT(Well Know Text) 格式，方便程式處理。

表2 輸入資料格式

```
POINT(x y)
POINT(250000,2300000)
POINT(250100,2300000)
POINT(250200,2300000)
...
```

3. 內插演算法設計

數學上的內插 (interpolation) 定義為：由已知的離散數據求未知數據的過程或方法。在空間分析上常見的內插法包括了：距離反比權重法(Inverse Distance Weight, IDW)、克利金法(Kriging)及雲形曲線法(Spline)。本研究採用距離反比權重法(Inverse Distance Weight, IDW)，利用已知雨量站位置推台灣地區其他位置的降雨情形。

IDW 公式如下(1)：

$$z_j = w_j \sum_{i=1}^N \frac{z_i}{d_{ij}^\alpha},$$

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

其中 Z_j 為待評估網格雨量值， Z_i 為已知雨量站降雨量， N 為內插過程中使用的已知雨量站數目， w_j 為權重， d_{ij} 為估計點與已知點的距離， $\alpha=2$ 。

為了讓內插運算可以平行分散處理，本研究依據 IDW 公式設計 Map 及 Reduce 函式演算法。在 Map 函式中讀取待評估網格位置，透過 IDW 公式進行評估運算，Reduce 函式則負責將網格位置整合後輸出至檔案上。

```
//*****
//map function
//*****
map(string key, string value)
//key: 待評估網格key
```

```
//value: 待評估網格坐標
begin
  Take the Evaluate Point, EvaluateP
  Find the nearest five rainfall stations, 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
  Emit ( key, EvaluateValue)
end

//*****
// reduce function
//*****
reduce(string key, Iterator value)
//key: cell key
//value: IDW 推估結果值
begin
  Emit(AsString(value))
end
```

4. 實驗結果

為了驗證平台執行情形，本研究將台灣地區網格解析度分別設定為 1000、100 及 50 公尺，藉此產生不同的檔案大小。台灣地區網格資料經由處理後，產生的資料量如表 3 所示。

表 3 網格資料量

網格數	解析度(m)	檔案大小
377*142	1000	1.7MB
3770*1420	100	173.9MB
7540*2840	50	695.4MB

為了測試在不同 Node 執行情形，實驗過程中分別模擬以 3 個、2 個及 1 個 Node 執行雲端處理程序，並觀察其執行情形。執行結果如圖 7 所示。

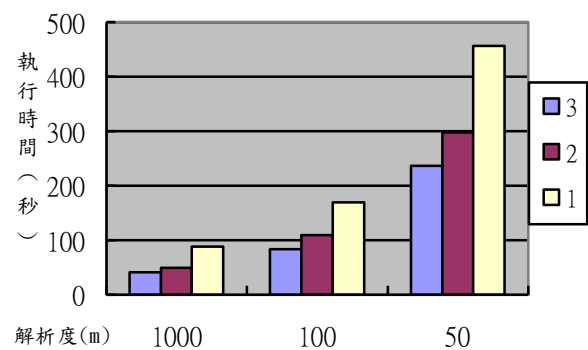


圖 7 實驗執行結果

由實驗結果可以觀察出運算 Node 數量會影響到運算所需要的時間，在網格解析度為 50 公尺時，3 個 Node 平行運算處理時間為 236 秒，如在僅有 1 個 Node 處理時則需花費 456 秒。由文獻回顧和與本研究實驗數據，可以驗證分散式平行處理

的確在巨量資料分析與處理上可以發揮其功用，當期望獲得較短的處理時間時，擴充運算 Node 可以獲得較短的運算時間。

七、結論

雲端運算已經是資訊產業重要的發展方向，全世界正在呼籲節能省碳的目標下，雲端運算技術可以有效節省伺服器機房建置量。本研究提出的空間雲端計算平台架構，透過 MapReduce 技術解決傳統 GIS 軟體無法處理的巨量資料的運算問題，同時雲端平台也扮演如同軟體元件銀行角色，後端有龐大的運算資源做為後盾，用戶端需要使用運算服務，可以在軟體元件銀行中找到需要的運算服務，達到資源共享的目的。

透過本研究的實驗可以看出平行分散處理可以有效縮短運算時間，運算節點愈多可以縮短資料運算所需要的時間。

參考文獻

- [1] Qichang Chen, Liqiang Wang and Zongbo Shang, "MRGIS: A MapReduce-Enabled High Performance Workflow System for GIS", *IEEE Fourth International Conference on eScienc3*, pp646-651, 2008.
- [2] Nathan Kerr, "Alternative Approaches to Parallel GIS Processing", M.S. Thesis, Kaiserslautern Tech. University, Germany, 2009.
- [3] 陳澄, "雲端策略—雲端運算與虛擬化技術", 天下雜誌社, 2010.
- [4] Jeffrey Dean and Sanjay Ghemawat, "Mapreduce: Simplified data processing on large clusters", *Commun.ACM*, vol. 51, no.1, pp. 137-150, 2008.
- [5] Web Processing Service, <http://www.opengeospatial.org/standards/wps>.