

# 建構雲端資源管理整合平台—以 Hadoop 為例

王淑卿<sup>1</sup>  
留旻懋<sup>1</sup>

李金鳳<sup>1</sup>  
洪瑋修<sup>1</sup>

江茂綸<sup>1\*</sup>  
簡輝銘<sup>1</sup>

曾莉雅<sup>2\*</sup>  
毛上仁<sup>1</sup>

<sup>1</sup>朝陽科技大學

<sup>2</sup>華夏技術學院

\*: 聯絡人

mlchiang@cyut.edu.tw; lytseng@cc.hwh.edu.tw

## 摘要

由於雲端技術的快速發展，為提供雲端運算服務一個穩定及較低成本的雲端平台，因此在資源分享的概念下，本研究建置一個能夠讓提供服務的供應商建立其提供服務的平台空間。因為 Hadoop 是目前最常見且實際運用在大規模商業環境上的雲端運算平台之一，強大而完整的基礎架構可以減少大量的雲端架構開發的時間，大量部署時也相當迅速，不但有許多重量級的雲端運算服務提供者正在使用及投入開發，也與 Google 的雲端環境相似，使 Hadoop 成為雲端服務開發的最佳平台。因此，在本研究中以 Hadoop 分散式系統為基礎，提出雲端資源管理架構，利用 Pool 進行資源的管理，並以此架構開發一套 Web 系統，讓使用者能更方便及快速得到雲端資源。本研究所建置的平台，可以依據服務提供者的需求，依其預算或資源需求給予適當的資源配置。

**關鍵詞：**雲端運算、Hadoop、Hbase、虛擬化、雲端資源管理。

## Abstract

Due to the rapid development of cloud technology, a stable and lower cost of the cloud platform is used to provide cloud computing services. Therefore, in the concept of resource sharing, a service platform that cloud service provider can support their services is constructed in this study. Hadoop is one of the most common and practical cloud computing platforms used in large-scale commercial environment. Hadoop has a powerful and complete infrastructure, deployed in large numbers fairly quickly, hence, can reduce the time of a large number of cloud-based developments. There are many heavyweight cloud computing services providers are using to invest in developing, also similar to the cloud environment of Google, Hadoop become the best

platform for the development of cloud services. Therefore, the Hadoop distributed systems is used to propose a cloud resources management structure in this study. In addition, the pool mechanism can support to manage the resources and develop a Web system, allowing users to be more convenient and fast to get cloud resources. In the proposed architecture, based on the needs of service providers, the appropriate allocations of resources are given.

**Keywords:** Cloud computing, Hadoop, Hbase, Virtualization, Cloud resource management.

## 1 前言

由於現今電腦網路頻寬及硬體設備相關技術的持續發展演進，使得網際網路的相關應用更加蓬勃發展。架構在網際網路的雲端運算 (Cloud Computing) 為一新興的分散式系統 (Distributed System) 之概念，其使用較低效能的主機來達到高可靠性及高效率的運算能力 [1,6]。由於雲端運算概念的興起，使得網際網路服務的應用，已透過雲端運算擴大其服務層面的廣度及服務內容的深度。

雲端運算不是一項全新技術，而是一項概念。雲端運算是一種分散式運算的概念，「雲」即為網際網路；「端」則是指使用者端 (Client) 或泛指使用者運用網路來完成服務。換言之，雲端運算是一種新興的、極具延展能力的運算方式，能把資訊科技功能，包括運算、儲存及頻寬，以「服務」的形式，透過網際網路提供給網路的使用者 [3,5]。雲端運算的精神是強調服務，並能依照使用者的需求提供客製化服務。雲端運算改變了傳統網路服務供應商的運作模式，創造了新型的服務方式，能夠把資訊科技的能力，包括運算能力、儲存能力以及頻寬速度，透過網際網路提供給使用者 [2,4]。

國家標準技術研究院(National Institute of Standards and Technology; NIST)將現今雲端運算技術所衍生以及存在的服務分為三種模式，通稱為 SPI 模式，分別為軟體(Software)、平台(Platform)以及基礎設施(Infrastructure)，各自的定義如下[6]：

#### **軟體即服務(Software as a Service ; SaaS)：**

在雲端環境中所提供的服務最為廣泛的即為 SaaS，SaaS 是透過網際網路在雲端環境中提供應用程式，以提供使用者進行服務的存取。狹義的 SaaS 是雲端運算環境中所提供的常見服務，包括：電子信箱、社交網路、網頁遊戲...等。但在雲端運算環境中所包含的範圍最為廣泛，像是提供給使用者的使用者介面、防火牆、虛擬機器...等，都是屬於 SaaS 所涵蓋的範圍。所以，若是沒有 SaaS 作為基礎，並扮演著中介的角色提供使用者便利的雲端運算環境，使用者將無法使用雲端運算環境的硬體設備，這樣雲端運算就僅是一般的大型運算中心。因此，在雲端運算的環境裡，SaaS 所涵蓋的範圍最大也是最重要的部分。

#### **平台即服務(Platform as a Service ; PaaS)：**

PaaS 是將使用者開發的應用程式或網頁部署在雲端運算環境上，使用者不須自己架設所需的開發平台環境，供應商將提供開發的環境、限定開發的語言、...等，使用者僅需遵循供應商所供應的平台就可以進行應用軟體的開發或網站的建置。亦即，PaaS 是 SaaS 衍生出來的一種服務方式，使用者不需要自己建構所需要的平台環境，利用供應商所提供的平台就可以進行網站的建置或應用軟體的開發，例如：Google App Engine、Microsoft Azure 等。

#### **基礎設施即服務(Infrastructure as a Service ; IaaS)：**

IaaS 提供雲端運算環境中的硬體設備資源讓使用者使用，有時可以稱為「公用運算」(Utility Computing)，也就是提供處理器、儲存硬碟、網路頻寬與其他資源的租用服務。換言之，IaaS 除了提供硬體設備資源，也提供了運算能力以及儲存能力。

由於雲端運算的特性，使得現今企業在雲端上提供的服務越來越多元化，例如 Google 的電子信箱、Dropbox 的儲存空間、Facebook 的交流空間、...等，甚至連網路遊戲都只需透過瀏覽器，這也使雲端環境的使用者快速的增加。換言之，只要使用者利用設備連上網際網

路，就可以使用雲端運算伺服器的服務及能力，且在雲端運算上的服務皆具有可攜的特性。亦即，在任何一部設備使用雲端運算的服務，都可以在另一部設備上繼續未完成的工作。

傳統的網際網路服務的型態為每一家服務供應商都擁有自己的伺服器，若是要服務更多的使用者則必須提升伺服器的能力或增加伺服器的數量，但當使用者的數目未達一定的程度時，則會浪費未使用的資源，以致造成企業成本的浪費。雲端運算利用了虛擬化(Virtualization)的技術來提供服務，將整個伺服器或個人電腦的資源虛擬成一個資源池(Pool)[18]，可以依照服務的需求或服務的種類來提供資源，並且可以結合工作排程(Job Schedule)的方式，來降低系統資源的浪費，達到快速回應以及減少等待時間的特性[2,5]。因此，一個穩定及較低成本的雲端平台，讓提供服務的供應商建立其提供服務的平台空間是值得建置的架構。因此，在本研究中以 Hadoop 分散式系統為基礎，提出雲端資源管理架構，並開發一套 Web 系統，讓使用者能更方便及快速得到雲端資源。

在本文第 2 節將說明本研究所採用的相關技術，第 3 節說明本研究所提出的技術應用，第四節則為系統建置與測試，第 5 節為結論與未來研究。

## **2 文獻探討**

在雲端運算環境的相關研究中，使用 MapReduce 演算法的 Hadoop 分散式系統，已被各大企業所廣泛採用，更是開發雲端運算平台的首選。本節將說明 Hadoop 分散式系統及 Hadoop 的 Hbase 分散式資料庫。

### **2.1 Hadoop 分散式系統**

Hadoop 是 Apache 軟體基金會所研發的開放源碼平行運算編程工具和分散式檔案系統 [7,16,17]，Hadoop 的原始作者是 Doug Cutting，最初是做為 Nutch 這個開放原始碼的搜尋引擎的一部份。Hadoop 是以 java 寫成，可以提供大量資料的分散式運算環境。簡單的說，Hadoop 就是建立一個 Cluster 平台，利用 MapReduce 的概念將一個工作分配到很多個 Cluster 平行運算。其中，Map 是將一個工作分配到多個運算節點(Node)執行，Reduce 則是將各個節點執行的結果重新整合成最後的結

果，而 Hadoop 就是提供一個雲端運算的平台。Hadoop 的特色包括[12,13]：

- 巨量：擁有儲存與處理大量資料的能力。
- 經濟：可以用在由一般個人電腦所架設的叢集環境之內，可以減少成本。
- 效率：藉由平行分散檔案的處理，得到快速的回應。
- 可靠：當某節點發生錯誤，系統能即時自動的取得備份資料以及佈署運算資源。

Hadoop的定位是用來處理與儲存大量資料的雲端運算平台，藉由平行分散檔案的處理得到快速的效率。在Hadoop的平台上，可運用一般的主機架設叢集環境，當某節點發生錯誤時，系統能及時自動的取得備份資料以及佈署運算資源，提供高容錯的能力。

Hadoop 架構包含兩大核心[9]：

- **MapReduce**：MapReduce 是一個分散式程式的框架，讓雲端服務開發者可以很簡單的撰寫程式，利用大量的運算資源，加速處理龐大的資料量。一個 MapReduce 的運算工作可以分成兩個部份，分別為 Map 和 Reduce。大量的資料在運算開始的時候，會被系統轉換成一組組(key, value)的序對，並自動切割成許多部份，分別傳給不同的 Mapper 來處理。待 Mapper 處理完成後，將運算結果整理成一組組(key, value)的序對，再傳給 Reducer 整合所有 Mapper 的結果。最後，才能將整體的結果輸出。MapReduce 程式流程如圖 1 所示。

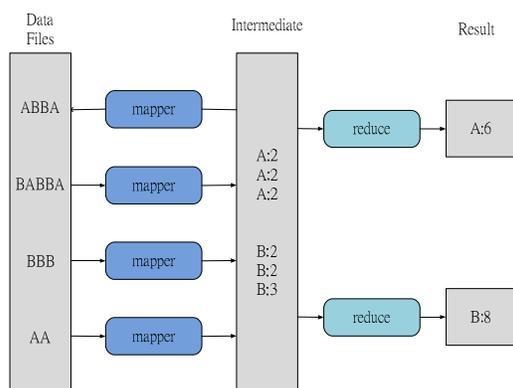


圖 1 MapReduce 程式流程

- **HDFS(Hadoop Distributed File System)**：HDFS 是 master/slave 架構，由兩種角色組成，分別為 Name node 及 Data nodes[8,15]。Name node 負責系統檔案中各個檔案屬性權限等資訊(metadata, namespace) 的管理及儲存，而 Data node 通常由數以百計的節點擔任。一個資料檔會被切割成數個較小的區塊(Block)儲存在不同的 Data node 上，每一個區塊還會有數份副本(Replica)存放在不同節點。透過區塊副本的儲存，當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，當有存取檔案的需求時，協調 Data node 負責回應。而當有節點損壞時，Name node 也會自動進行資料的搬遷和複製。HDFS 的架構如圖 2 所示。

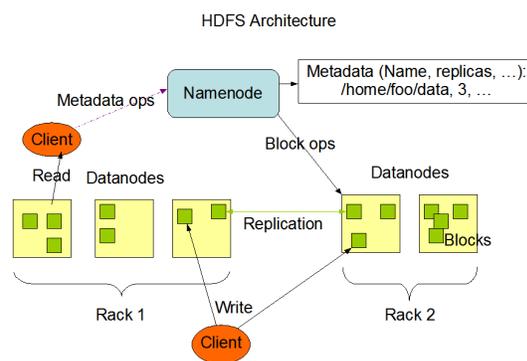


圖 2 HDFS 架構

## 2.2 Hbase 分散式資料庫

Hbase 是一個分散式開源資料庫，基於 Hadoop 分散式文件系統，模仿並提供了基於 Google 文件系統的 Bigtable 資料庫的所有功能 [10,11]。整體而言，HBase 是一個開放源碼的高可靠性、高性能、可伸縮、非關聯的分散式資料庫，用以儲存大規模結構化數據。Hbase 的目標是處理非常龐大的表，可以用普通的電腦處理超過 10 億行資料，並且有數百萬列元素組成的資料表。

HBase 使用和 Bigtable 非常相似的資料模型儲存大規模結構化數據，是一個可以快速存取大量資料的一個資料庫設計。HBase 利用 Row Oriented 以及 Column Family 的概念，方便存入具有不同 Column 屬性的 Data 以及根據 Row Key 快速拿出想要的 Data。HBase Table 如表 1 所示。

表 1 HBase Table

Row	column-family1		column-family2		
Key	column1	column2	column1	column2	column3
key1	t1:abc		t4:dfads		
	t2:gdxdf		t3:hello		
key2	t3:abc		t4:dfads		t2:dfdsfa
	t1:gdxdf		t3:hello		t3:dfdf

其中，

Row Key：Table 的主鍵，Table 中的記錄依照 Row Key 排序。

Column Family：列族，Table 在水平方向有一個或者多個 Column Family 組成。一個 Column Family 中可以由任意多個 Column 組成，即 Column Family 支持動態擴展，無需預先定義 Column 的數量以及類型，所有 Column 均以二進制格式儲存，用戶需要自行進行類型轉換。

Hbase 運行方法：

- **Table & Region**：當 Table 隨著所記錄的數量不斷增加而變大後，會逐漸分裂成多份 Splits，成為 Regions。一個 Region 由 (startkey, endkey) 表示，不同的 Region 會被 Master 分配給相對應的 RegionServer 進行管理，如圖 3 所示。

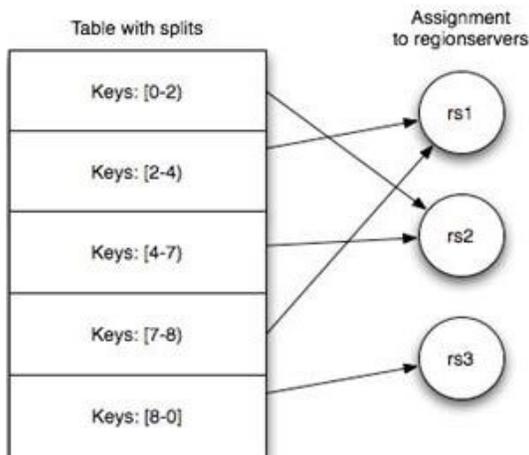


圖 3 Table & Region 示意圖

- **ROOT & META Table**：HBase 中有兩張特殊的 Table，包括 ROOT 和 META，如圖 4 所示。

其中，

META：記錄用戶表的 Region 信息，META 可以有 multiple Region。

ROOT：記錄 META 表的 Region 信息，ROOT 只有一個 Region。

zookeeper 中，則記錄 ROOT 表的位置。

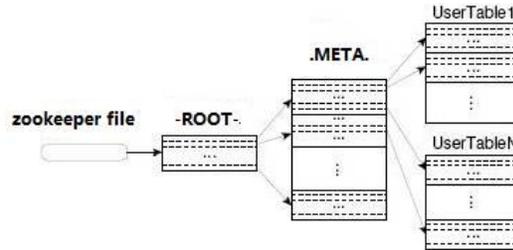


圖 4 ROOT & META 示意圖

Client 使用用戶數據之前需先訪問 zookeeper，然後使用 ROOT 表，接著使用 META 表，最後才能找到用戶數據的位置去擷取所要的資訊。由於中間需要進行多次網絡操作，因此於 Client 端以 Cache 進行緩衝儲存。

### 3 雲端資源管理架構

雲端運算服務為企業經營帶來許多新觀念的改變，透過雲端資源的分享，企業不必再投入資訊建置的成本或人力的需求，亦無需為短暫時間內大量的網路需求量而購買昂貴的網路設備。雲端運算並不是在技術上的創新，而是一個創新的概念，其轉換服務的方式以及商業營運的模式。而隨著現今快速的資訊科技及網路技術的發展，以使用者為導向的雲端運算概念，成為了各界矚目的焦點。雲端運算的相關服務是由網路、運算、及服務這三類組成的。亦即，雲端運算提供的服務都是透過網路連線，由在網路的另一端所提供強大的運算能力以及其他服務，讓個人或企業藉由改變過去的資料處理方式，來提高作業的效率，也可以省下大量的成本，進而同時獲得資訊科技的創新應用。

### 3.1 系統架構

本研究之所建構的系統，首先在每台個人 PC 上安裝 Ubuntu 作業系統，在內部安裝及設置 Hadoop 和 Hbase，並設定其中一台主機為 Namenode，將其他主機設定為 Datanode。最後，將各個主機連結為叢集式雲端運算平台環境。圖 5 所示為本研究所建構的系統之概念圖。

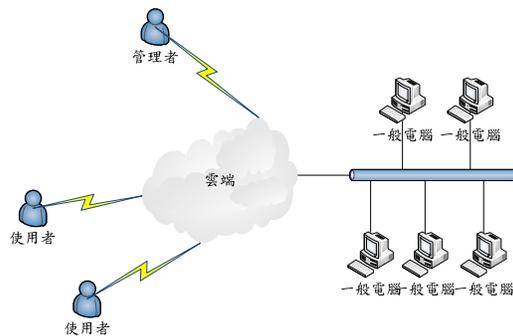


圖 5 系統概念圖

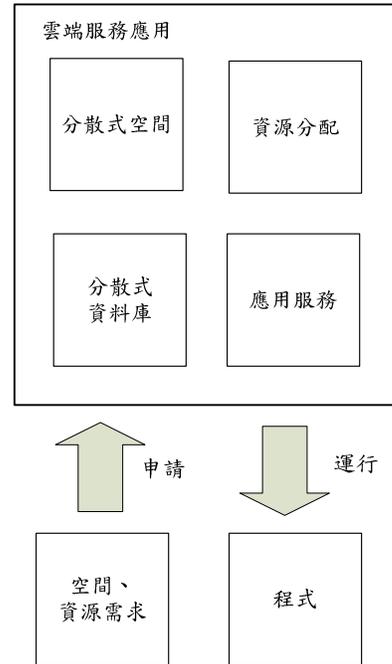


圖 6、系統架構圖

在系統架構中主要分為四大元件，包含分散式硬碟管理、資源分配、應用服務、及分散式資料庫。

- (1) 分散式硬碟管理之主要功能為整合所有節點的雲端空間，並依照權限進行空間的切割與限制。
- (2) 資源分配將所有節點資源整合成一個大資源池，再進行相對應配置，將其分成許多個小資源池。
- (3) 應用服務則可依不同程式的類型運行服務的功能。
- (4) 運作流程將使用者送出的申請需求記載於分散式資料庫中，管理者依其需求給予相對應的硬碟空間、以及資源的配置，使用者立即可獲得運行應用服務之功能。

本研究所建置的系統，其系統架構如圖 6 所示。

### 3.2 分散式硬碟管理

HDFS 提供傳統階層式型的文件組織，與大多數其他文件系統類似，用戶透過指令可以創建目錄，並在其間創建、刪除、移動和重命名文件。在本研究所建構的系統中，經由簡化其操作指令，並針對各使用者進行權限切割，同時提供透過 web 端輕鬆管理雲端硬碟的人性化介面。

相關設定指令：

chmod：改變文件的權限。

使用方法：hadoop fs -chmod [-R] <MODE[,MODE]... | OCTALMODE> URI [URI ...]

chown：改變檔案的擁有者。

使用方法：hadoop fs -chown [-R] [OWNER][:[GROUP]] URI [URI]

cp：將檔案從原路徑複製到目標路徑。

使用方法：hadoop fs -cp URI [URI ...] <dest>

get：複製檔案到本地文件系統。

使用方法：hadoop fs -get [-ignorecrc] [-crc] <src> <localdst> ls

mkdir：創建檔案目錄。

使用方法：hadoop fs -mkdir <paths>

mv：將檔案從原路徑移動到目標路徑。

使用方法：hadoop fs -mv URI [URI ...] <dest>

put：從本地檔案系統中複製單個或多個原路徑到目標檔案系統。

使用方法：hadoop fs -put <localsrc> ... <dst>

rm：刪除指定的檔案。

使用方法：hadoop fs -rm URI [URI ...]

rmdir：刪除路徑及路徑下的檔案。

使用方法：hadoop fs -rmdir URI [URI ...]

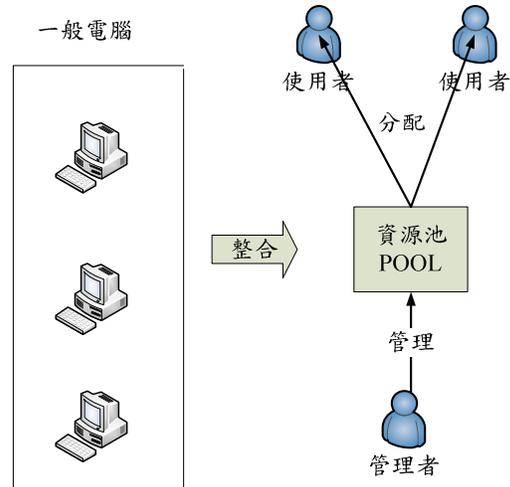


圖 8 資源分配架構

HADOOP 整合各節點的硬體資源，建置 HDFS 分散式檔案系統，透過 Hadoop:50070 之資訊畫面，則可觀看目前 HDFS 使用的資訊，如圖 7 所示。

**NameNode 'hadoop1:9000'**

Started: Wed Nov 14 15:39:34 CST 2012  
Version: 0.20.2, r911707  
Compiled: Fri Feb 19 08:07:34 UTC 2010 by chrisdo  
Upgrades: There are no upgrades in progress.

[Browse the filesystem](#)  
[Namenode Logs](#)

**Cluster Summary**

667 files and directories, 352 blocks = 1019 total. Heap Size is 6.06 MB / 963 MB (0%)

Configured Capacity	: 586.38 GB
DFS Used	: 926.47 MB
Non DFS Used	: 68.83 GB
DFS Remaining	: 516.64 GB
DFS Used%	: 0.15 %
DFS Remaining%	: 88.11 %
Live Nodes	: 3
Dead Nodes	: 0

**NameNode Storage:**

Storage Directory	Type	State
/var/hadoop/hadoop-hadooper/dfs/name	IMAGE_AND_EDITS	Active

[Hadoop](#), 2012.

圖 7 HDFS 資訊畫面

### 3.3 資源分配

本研究使用 POOL 資源池觀念對雲端資源進行整合，並分配給不同需求者，如圖 8 所示。

建立 POOL.XML 與 HADOOP 相關建置文檔的 mapred-site.xml[14,18]，來進行資源分配之動作，在 mapred-site.xml 添加下列程式：

```
<property>  
<name>mapred.fairscheduler.allocation.file</name>  
<value>/home/hadooper/hadoop/conf/pool.xml</value>  
</property>
```

讓 mapred-site.xml 與 POOL.xml 進行串聯，並在 POOL.xml 依據使用需求填入以下標籤：

```
<poolname="my_queue">  
<minMaps>10</minMaps>  
<minReduces>10</minReduces>  
<maxRunningJobs>10</maxRunningJobs> <minSharePreemptionTimeout>300</minSharePreemptionTimeout>  
<weight>2.0</weight>  
</pool>  
<username="my_queue ">  
<maxRunningJobs>5</maxRunningJobs>  
</user>
```

標籤 pool 代表的是在資源池的分配量，所需求的資源數量，是經由標籤內容來設定。其中標籤中的 user 代表在這資源池的使用者，透

過將 pool 與 user 兩個標籤中的變數 name 命名為相同名稱,即可將 POOL 分配到對應使用者。

minMaps: 該 pool 可使用的 map slot 數

minReduces: 該 pool 可使用的 reduce slot 數

maxRunningJobs: 該 pool 最大運行作業數

mapred.job.queue.name 讓 hadoop 得知 Job 要運行在哪一個 pool 當中。

### 3.4 分散式資料庫

本研究分散式資料庫採用 HBASE, 在雲端使用者部分, 以 user 帳號作為 RowKey, 將使用者個人資訊及分配資源數量分別記載在 user & pool 兩個 Column Family 當中。為了讓 WEB 系統端能夠存取雲端硬碟(HDFS), 透過 HBASE 以及 Shell Script 同步進行資料傳輸。一邊以檔案名稱為 Row Key 並以 user 身分做權限區分, 將其檔案資訊記載入分散式資料庫當中。一邊則將其檔案上傳至雲端硬碟(HDFS), 透過分散式資料庫來幫助 WEB 介面存取雲端資源相關資訊。

## 4 系統功能測試

為驗證本研究所建構的系統架構, 確實可以實現讓使用者獲取雲端運算資源之相關管理。因此本研究實際開發系統, 使用 vmware esxi 建置虛擬化環境, 叢集多台 PC, 並透過 Hadoop 整合節點。經由 WEB 介面進行 POOL 資源池切割、雲端硬碟分配、雲端運算資源提供, 本研究實驗環境如下:

虛擬化系統: VMWare ESXI 5.1

作業系統: Ubuntu10.04

開發工具: 記事本、Adobe Dreamweaver CS5

資料庫軟體: Hbase 0.20.2

雲端平台: Hadoop 0.20.2

開發語言: PHP5、ShellScript、XML、JavaScript、JAVA、C++

本研究透過 WEB 介面, 在不同的作業系統只要有上網存取能力, 都能讓使用者操作其相關雲端資源。

### 4.1 雲端空間、資源申請及分配

使用者在運行服務前, 需先申請雲端空間以及資源。透過網頁申請頁面, 本系統提供一套資源申請評估依據, 做為申請的填寫標準, 使用者可簡單填寫並送出資料表單, 如圖 9 所示。其中, 1 台 VM 建議申請容量為 10G, 但使用者可依情況調整申請。



圖 9 資源申請

當系統收到使用者申請的表單後, 經過自動換算申請數量, 即可動態進行相關資源配置, 分配資源給予使用者, 如圖 10 所示。

分配換算公式(在 1 台 VM 容量為 10G 時)如下:

$$\text{MAP} = (\text{申請容量} * 1024) / \text{HDFS Size}(64\text{MB})$$

$$\text{Reduce} = \text{MAP} * 50\%$$

$$\text{Running Job} = \text{MAP} * 25\%$$

$$\text{MAX Running Job} = \text{Reduce}$$



圖 10 POOL 分配功能

## 4.2 服務的運行測試

使用者可利用服務上傳的功能上傳欲於雲端空間中執行的程式，也可利用下載、刪除功能進行檔案之管理，如圖 11 所示。檔案上傳的格式與規則規定如下：

1. 開放 JAVA、PHP、C++ 等程式。
2. JAVA 程式：需打包成 jar 包，並設置 jar 包之 Main-Class 為所要執行的 Class。
3. C++ 程式：需打包成 CPP 格式。
4. 將 JAVA 改寫成 Hadoop Map/Reduce。

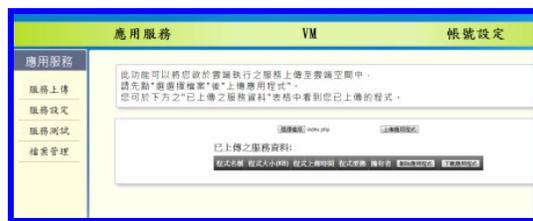


圖 11 服務上傳功能

當使用者上傳所欲提供的服務後，可應用服務執行運行上傳之服務。如圖 12 所示即為 WordCount 程式運行的結果。



圖 12 服務運行狀態

## 5 結論與未來研究

在本研究中建構了雲端資源管理整合平台，此平台是以 Hadoop 分散式系統為基礎並加以延伸發展而成。透過 POOL 資源池觀念將雲端資源進行整合管理，並依據使用者需求進行資源的分配。透過資源池之資源與 HDFS 整合，達成運算及儲存空間的虛擬機器 (Virtual Machine)。本研究透過系統的實際開發，建置

雲端資源管理的整合平台，使得系統管理者與使用者，能更方便的進行系統操作，不再需要輸入繁瑣的指令集，也能使用或者管理雲端運算環境中的龐大資源。

目前本研究所架構的系統為半動態之設計，因此未來規劃持續建構的項目如下：

- (1) 全動態的 POOL 分配：透過更方便的 WEB 操作來減輕管理者所需操作之項目。
- (2) 使用者介面優化：在使用者介面上仍有許多不足及尚未完善部分，為了讓所發展之系統能更為便捷，規劃將優化使用者介面。

## 參考文獻

- [1] Aymerich, F.M., Fenu, G. and Surcis, S., "An Approach to A Cloud Computing Network," *Proceedings of 1st International Conference on the Applications of Digital Information and Web Technologies*, pp. 113-118, August 4-6, 2008.
- [2] Birman, K., Chockler, G. and Renesse, R., "Toward A Cloud Computing Research Agenda," *ACM SIGACT News*, Vol. 40, No. 2, pp. 68-80, June 2009.
- [3] Gong, C., Liu, J., Zhang, Q., Chen, H. and Gong, Z., "The Characteristics of Cloud Computing," *Proceedings of the Parallel Proceeding Workshops*, 2010.
- [4] Grossman, R.L., Gu, Y.H. and Sabala, M. and Zhang, W.Z., "Compute and Storage Clouds Using Wide Area High Performance Networks," *Future Generation Computer Systems*, Vol. 25, No. 2, pp. 179-183, 2009.
- [5] Zhang, S., Zhang, S., Chen, X. and Huo, X., "Cloud Computing Research and Development Trend," *Proceedings of the Second International Conference on Future Networks*, pp. 93-97, 2010.
- [6] 黃重憲, "淺談雲端運算," [http://www.cc.ntu.edu.tw/chinese/epaper/0008/20090320\\_8008.htm](http://www.cc.ntu.edu.tw/chinese/epaper/0008/20090320_8008.htm).
- [7] 國網中心 HADOOP 環境安裝教學, [http://trac.nchc.org.tw/cloud/wiki/Hadoop\\_Lab7](http://trac.nchc.org.tw/cloud/wiki/Hadoop_Lab7)
- [8] 國網中心 HDFS SHELL 指令, [http://trac.nchc.org.tw/cloud/wiki/Hadoop\\_Lab2](http://trac.nchc.org.tw/cloud/wiki/Hadoop_Lab2)
- [9] 國網中心 MAP/REDUCE 程式開發教學, [http://trac.nchc.org.tw/cloud/wiki/Hadoop\\_Lab5](http://trac.nchc.org.tw/cloud/wiki/Hadoop_Lab5)

- [10] 國網中心 Hbase+thrift+php 教學手冊, <http://trac.nchc.org.tw/cloud/wiki/waue/2010/HbaseThrift>
- [11] HBASE 安裝教學, <https://sites.google.com/site/waue0920/Home/hbase/hbase-cong-ji-an-zhuang>
- [12] 雲端論壇(台灣), <http://forum.hadoop.tw/>
- [13] HADOOP 技術論壇(大陸), <http://www.hadoopor.com/index.php>
- [14] HADOOP API, <http://hadoop.apache.org/docs/r0.20.2/api/>
- [15] HADOOP HDFS 官方操作指南, [http://hadoop.apache.org/docs/r0.20.2/hdfs\\_shell.html](http://hadoop.apache.org/docs/r0.20.2/hdfs_shell.html)
- [16] HADOOP 官方權限指南, [http://hadoop.apache.org/docs/r0.20.2/hdfs\\_permissions\\_guide.html](http://hadoop.apache.org/docs/r0.20.2/hdfs_permissions_guide.html)
- [17] HADOOP FairScheduler 官方指南, [http://hadoop.apache.org/docs/r0.20.2/fair\\_scheduler.html](http://hadoop.apache.org/docs/r0.20.2/fair_scheduler.html)
- [18] HADOOP POOL 切割, <http://dongxicheng.org/mapreduce/hadoop-permission-management/>