

# Session High-Availability and Load Balancing in Virtual Application Delivery System

## 虛擬應用交付系統中的 Session 高可用性與負載平衡

毛奕翔

財團法人資訊工業策進會

雲端系統軟體研究所 雲端鉅資研發中心

patrickmou@iii.org.tw

### 摘要

虛擬桌面基礎架構 (Virtual Desktop Infrastructure, VDI) 是目前雲端運算主要應用之一，資策會的產品 PC<sup>2</sup>(Personal Cloud Computer)是 VDI 的實作，提供跨平台遠端桌面服務，使用 Session 作為連線的方式，我們提出一個架構可以在此產品上建立出一個中央管理閘(Session Manager Gateway)，並由此中央管理閘提供集中式管理架構，以達到服務的高可用性，並透過與伺服器的資料庫連線取得伺服器負載資料達到負載平衡功能。

**關鍵詞：**雲端運算、遠端桌面、負載平衡、高可用性

### Abstract

Virtual Desktop Infrastructure, VDI, is the main stream in the cloud computing appliance. PC<sup>2</sup> (Personal Cloud Computer) is an implementation of VDI serving cross platform remote desktop service. Using session as the connection method, the PC<sup>2</sup> service is constructed under the architecture of Session Manager Gateway. The system supports centralized management structure in able to achieve high-availability service. It also supports load balance by accessing the database to retrieve the servers' status.

**Keywords:** Cloud Computing, Remote Desktop, Loading Balance, High Availability.

### 一、前言

虛擬桌面基礎架構 (VDI) 是指託管操作系統和應用程序服務器，只要該設備可以連接到主機服務器，用戶可以從任何設備訪問虛擬化的操作系統和應用程序，透過虛擬化的方式可以在不同的操作系統，如 Linux、Ubuntu、Windows 7 上運行的設備進行操作。

目前資策會的相關產品 PC<sup>2</sup>，為雲端桌面應用服務是此應用的實作，提供簡化桌面管理和任務與提供集中性的安全性和數據保護，而集中管理可以降低運營成本，用戶端可以很容易地擴展、啟動與快速及高效地運行，並涵蓋「跨平台應用程式管理」與「個人化表單訂製」，統整使用者所需的應用程式清單列表與結合伺服器、網路、儲存、雲端桌面應用程式等，達到提升使用的便捷性與強化桌面高度整合的目標，使用者畫面如圖 1，系統管理者服務管理頁面如圖 2。

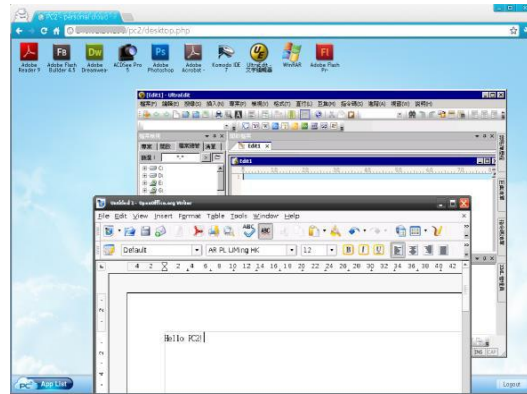


圖 1 PC<sup>2</sup> 使用者桌面截圖



圖 2 PC<sup>2</sup> 系統管理者服務管理頁面

叢集管理架構係為支援整體雲端虛擬桌面高可用性的一環，主要是為了實現雲端虛擬桌面緊急狀況發生時，可以由另一台備援設備啟動叢集管理技術，達成 PC<sup>2</sup> 有效持續運作不中斷特性，當有一方中斷時，另一方可在時間內，啟動高可用性機制 (High Availability / HA)，確保管控過程不中斷。

另一項則為負載平衡機制(Load Balance / LB)，系統資源的管理中，負載平衡占有重要的關鍵地位，當不同的伺服器共同提供服務給使用者時，有效的負載平衡機制，可更大地發揮及有效節省相關資源。例如 PC<sup>2</sup> 的負載平衡，經由增加一個控制閘(Gateway)控管與取得不同 PC<sup>2</sup> 伺服器的負載平衡資訊，並分配告知使用者該由何處進入，有效達成資料流控管與規劃，將資源負擔平均配置，有效發揮負載平衡強大的功能。

## 二、相關研究

在本文中實作在資策會的產品 PC<sup>2</sup> 上，因此首先說明 PC<sup>2</sup> 結構與相關使用技術，而其結構如圖 3，PC<sup>2</sup> 產品上主要有兩種構件，中央管理端(Session Manager)與應用伺服器(Application Server / AP Server)，中央管理端負責與使用者的溝通、狀態的監控、應用伺服器的分配等；應用伺服器則是應用的提供，如 Ubuntu 上的 Open Office 或 Windows 上的 Microsoft Office，為了提供跨平台的服務，在一台完整的 PC<sup>2</sup> 伺服器上會連接兩台不同平台的應用伺服器(Ubuntu、Windows)，但最低只需要一台 Ubuntu 之應用伺服器即可運作。

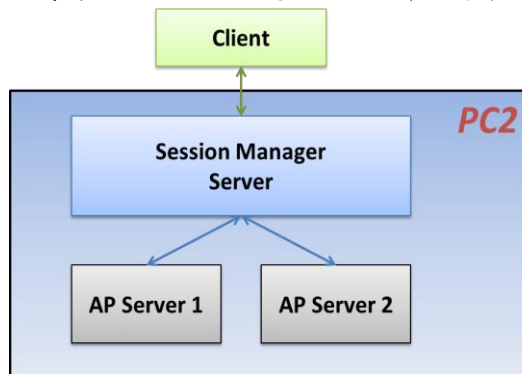


圖 3 PC<sup>2</sup> 架構圖

使用者利用瀏覽器(Chrom、Firefox)與中央管理端連線從而建立起 PC<sup>2</sup> 的應用服務，期間是以 Session 作為溝通的方式，但單一 PC<sup>2</sup> 伺服器的承載是有限的且一旦損毀則服務就需中斷，因此考量在多台 PC<sup>2</sup> 伺服器的環境下，建立叢集管理架構，使得能達成可用性與負載平衡成為本文的主要目標。

分析 PC<sup>2</sup> 伺服器的負載，需對實作方式做進一步的了解，資策會內對於 PC<sup>2</sup> 產品負載平衡的相關研究 [1]，以下介紹 PC<sup>2</sup> 的主要實作方式：

- PC<sup>2</sup> 提供了雲端桌面服務，主要為 RDP(Remote Desktop Protocol)[2][3] 的應用，使用主從式架構(Client-Server)[4]，

能將遠端應用伺服器的服務提供給使用者，其中包含畫面、聲音。

- 使用 eFinder[5]作為操作畫面的主體，提供給使用者圖形化介面，並使用 WebDAV[6]來連接對檔案執行操作。

有關可用性的研究中，有許多研究建立高效率的管理閘，並建構出錯誤回復的機制，如 [7]；有關負載平衡的研究中，[8]提出了一個雲端服務的三層架構，[9] [10]提出選擇伺服器的演算法。

## 三、研究方法

中央管理端(Session Manager)控管各個 Session 的進出與連結情形如圖 4 所示，為能防止當其中一台中央管理端控管失靈時，PC<sup>2</sup> 伺服器依然可順利運行維持高可用性(High Availability / HA)，以及能具有負載平衡(Load Balance)機制能針對中央管理端的負載平衡狀況，進行了解與協調，所以加入中央管理閘(Session Manager Gateway)，以期達成更順暢的界接效能。

為了記錄叢集所屬的 PC<sup>2</sup> 伺服器端點，在中央管理閘(Session Manager Gateway)建立簡單的資料庫表格，並儲存所需的 PC<sup>2</sup> 伺服器資訊，如編號、檢查時間、IP 位址等，在文中稱為伺服器狀態列表(Server Status List)。

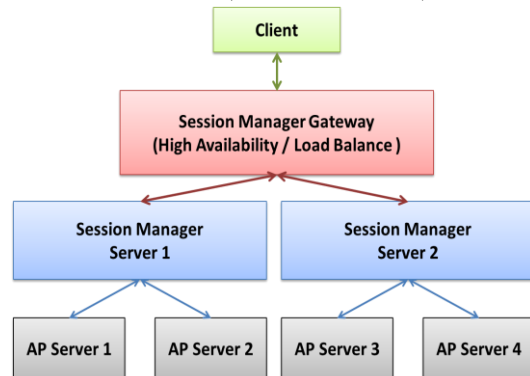


圖 4 中央管理閘高可用性與負載平衡架構圖

### (一) 負載平衡(Load Balance)機制

每一台 PC<sup>2</sup> 伺服器有自我監控的機制，這個機制會每 30 秒將伺服器的相關資訊記錄到資料庫中，此資訊包含所屬的應用伺服器(Application Server)資訊。

因此只需要在中央管理閘建立一個不中斷的循環，每次的循環與 PC<sup>2</sup> 伺服器自我監控的週期相同，週期內每一次會先讀取伺服器狀態列表中的各伺服器 IP 資訊，並依序與每一台伺服器連線，連線的內容是到各 PC<sup>2</sup> 伺服器

中的資料庫讀取所屬的狀態資料，即可以由每台 PC<sup>2</sup> 伺服器的資料庫中得知所需資訊，接著透過公式計算每台 PC<sup>2</sup> 伺服器的負載平衡分配優先權重(Priority)，最後儲存回伺服器狀態列表，其機制如圖 5。

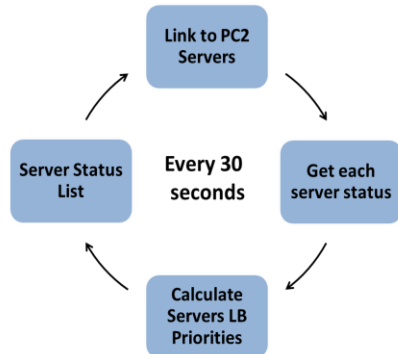


圖 5 中央負載平衡(Load Balance)機制

計算每台 PC<sup>2</sup> 伺服器的負載平衡分配優先權重(Priority)，採用 [1] 中的公式並稍作修改，將 RAM 的計算轉換為剩餘的百分比，使用計算方式如下：

$$\begin{aligned}
 Priority(x) = & \\
 & \{[100 - cpu_c(x)] \cdot W_{CPU}\} + \\
 & \{[ram_m(x) - ram_c(x)] / ram_m(x) \cdot W_{RAM}\} + \\
 & \{[sessions_m(x) - sessions_c(x)] \cdot W_{sessions}\}, \\
 \forall x: x \in servers & \quad (1)
 \end{aligned}$$

其中  $x$  為伺服器狀態列表中某一伺服器；在伺服器列表中會記錄  $cpu_c(x)$ 、 $ram_m(x)$ 、 $ram_c(x)$ 、 $sessions_m(x)$ 、 $sessions_c(x)$ ，分別代表該伺服器目前的 cpu 使用率、ram 最大空間、ram 目前使用空間、session 最大允許連線數、session 目前的連線數，而這些數值分別由各 PC<sup>2</sup> 伺服器讀取設定值或自動計算；相關權重值  $W_{CPU}$ 、 $W_{RAM}$ 、 $W_{sessions}$  則依照產品實際測試經驗設定。

在計算過程中若無法連線到某台 PC<sup>2</sup> 伺服器或是在取得的資訊中得知所屬應用伺服器無法使用，則將該 PC<sup>2</sup> 伺服器的負載平衡分配優先權重(Priority)設定為負數。

最後依照負載平衡演算法計算之結果儲存到伺服器狀態列表，使得使用者連線到中央管理端能依其中的負載平衡分配優先權重(Priority)得知各 PC<sup>2</sup> 伺服器的負載，進而分配適當的 PC<sup>2</sup> 伺服器使用。

## (二) 高可用性(High Availability / HA)機制

在上一階段的負載平衡機制中除了達成負載平衡的運算外，因為 PC<sup>2</sup> 伺服器不可用則負載平衡分配優先權重(Priority)為負數，同時也測試了 PC<sup>2</sup> 伺服器的可用性，但因為 30 秒的週期太長，在此週期內 PC<sup>2</sup> 伺服器也有損毀的可能，導致使用者被分配損毀的 PC<sup>2</sup> 伺服器，因此在使用者登入時需要做另一次檢查，以確保使用者能得到可用的 PC<sup>2</sup> 伺服器，而由於使用者登入的數量可能會很密集，這個檢查需要快速且對中央管理端產生的負載較低，此檢查實作流程如圖 6。

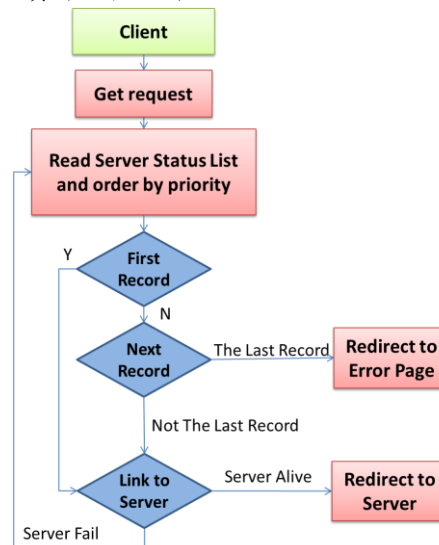


圖 6 中央管理端資料流程圖

使用者在連線時會在中央管理端產生一個連線的請求，當中央管理端收到此請求後首先會讀取伺服器狀態列表，將每個 PC<sup>2</sup> 伺服器 IP 位址與計算得到的 Priority 取出，並在同時依照各 PC<sup>2</sup> 伺服器的 Priority 由大到小依序排列，且將 Priority 為負數的 PC<sup>2</sup> 伺服器排除以避免使用已經損毀的 PC<sup>2</sup> 伺服器，儲存成排序伺服器列表。

接著由排序伺服器列表中取出第一個 PC<sup>2</sup> 伺服器 IP 位址，並嘗試進行連線，用以確定 PC<sup>2</sup> 伺服器可用，若可用則將使用者連線重新導向到此 PC<sup>2</sup> 伺服器進行服務，若不可用則讀取排序伺服器列表中的下一個 PC<sup>2</sup> 伺服器，並重複步驟以找到一台可用的 PC<sup>2</sup> 伺服器，如果排序伺服器列表中最後一個 PC<sup>2</sup> 伺服器都不可用，則會將使用者重新導向提示錯誤的網頁，以便使用者可以通知系統管理員修復 PC<sup>2</sup> 伺服器。

經由負載平衡(Load Balance)機制可以檢查各個主要元件的可用與否，包含網路、資料庫、中央管理端(Session Manager)與應用伺服器(Application Server)，若某一中央管理端

(Session Manager)失效則可將使用者導向另一台可用的 PC<sup>2</sup> 伺服器，達成高可用性(High Availability / HA)的目的。

#### 四、研究結果

依據以上架構實作實驗環境如圖 7，主機環境資訊如下：

- CPU: Intel Xeon E3 3.30GHz (8 核)
- RAM: 32G
- OS: Windows 7

在此機器上安裝 VirtualBox 以模擬 3 台 PC<sup>2</sup> 伺服器與 1 台中央管理閘(Session Manager Gateway)，每台模擬機器規格如下：

- CPU: 1 核
- RAM: 1G
- DISK: 50G
- 網路: 橋接介面卡(Bridging)
- OS: Ubuntu 10.04

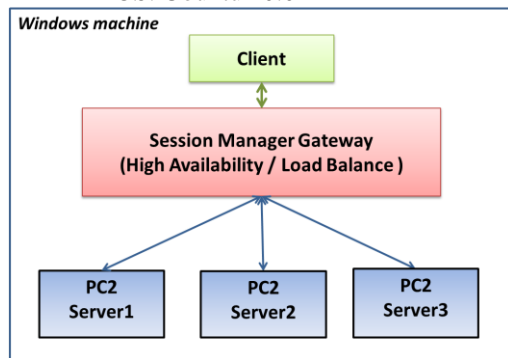


圖 7 實驗環境架構

在主機上產生機制，模擬使用者(Client)行為，每 0.5 秒會連線到中央管理閘(Session Manager Gateway)以測試是否正確執行，並在中央管理閘(Session Manager Gateway)中記錄分配的 PC<sup>2</sup> Server IP 以作為驗證依據，並將 3 台伺服器依照所有組合關機與開機，模擬各台伺服器錯誤可能產生的中斷，且以人工額外登入各台伺服器，使伺服器使用的 Session、CPU 與 RAM 產生變動，用於驗證負載平衡(Load Balance)機制，實驗過程中測試超過 5 萬次 Client 連線結果皆為正確，其分配比率如圖 8，其中 Server 3 分配比率較高的原因是在在三台伺服器情況結相同的情況下(無 session 登入、CPU 與 RAM 剩餘比率相同)，因為 IP 數值較小而被資料庫優先取用。

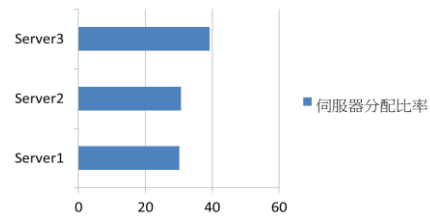


圖 8 伺服器分配比率

為了統計使用在中央管理閘(Session Manager Gateway)上使用的資源，使用 Bash 每 1 秒循環記錄系統資源使用狀態，其結果如圖 9 與圖 10，圖表中 CPU 與 RAM 的使用率變動與操作系統(Ubuntu 10.04)之行為有直接的關聯，但可以得知所使用的系統資源極低。

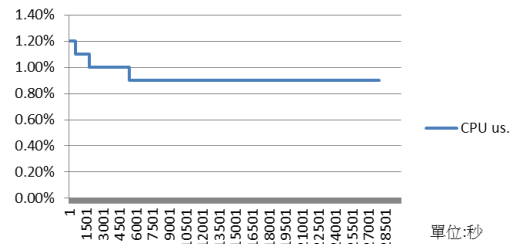


圖 9 CPU 使用率

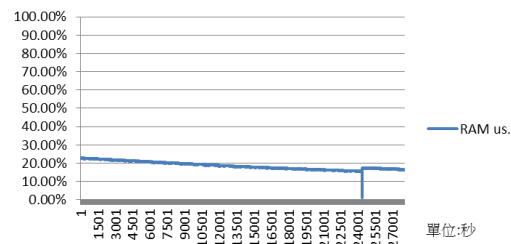


圖 10 RAM 使用率

#### 五、結論與討論

在虛擬桌面基礎架構(VDI)的環境下，叢集管理架構是提供了可用性的重要方式，可以達到服務的不中斷與集中式的控管，以利於服務供應商的營運；而當不同的伺服器共同提供服務給使用者時，有效的負載平衡機制，使伺服器等資源可以更效率的發揮，更大限度地發揮及節省相關資源。

在提出的架構下可以輕易簡單的架構出叢集管理系統，達到簡單的擴充與刪減伺服器，達到更高的管理性，並且與服務本身的相依性很低，最低只需伺服器使用的 Session、CPU 與 RAM 三個變數存入資料庫即可以達到目的，若需要更高要求的檢測只需要將服務或伺服器更多檢測結果寫入資料庫，在 Session Manager Gateway 提取並做判斷，就可達到更多的檢查。

在產品 PC<sup>2</sup> 中使用 RDP 來提供遠端桌面

服務，在此情況下伺服器還有許多負載的因素，如硬碟 IO 與網路速度等，在此次實驗中是使用 Session 的數量來簡化，對於真實情況還有需要改進的部分，另外還有多個 PC<sup>2</sup> 伺服器中應用伺服器、資料庫與檔案伺服器的共用問題，將是未來研究的目標。

## 六、致謝

本研究依經濟部補助財團法人資訊工業策進會「103 年度雲端運算系統及軟體技術研發計畫(3/3)」辦理。

## 參考文獻

- [1] 湯凱歲, “雲端虛擬桌面系統之分散式架構及負載平衡”, 臺灣網際網路研討會, pp. 193-197, 十月, 2013.
- [2] Microsoft, (2010). “White Paper: RDP Features and Performance,” [Online]. Available: <http://www.microsoft.com/en-us/download/details.aspx?id=31070>
- [3] Wikipedia, (2014). “Remote Desktop Protocol,” [Online]. Available: [http://en.wikipedia.org/wiki/Remote\\_Desktop\\_Protocol](http://en.wikipedia.org/wiki/Remote_Desktop_Protocol)
- [4] Wikipedia, (2014). “Client-Server model,” [Online]. Available: [http://en.wikipedia.org/wiki/Client-server\\_model](http://en.wikipedia.org/wiki/Client-server_model)
- [5] eFinder, (2013). “File manager for web,” [Online]. Available: <http://elfinder.org/>
- [6] Wikipedia, (2014). “WebDAV,” [Online]. Available: <http://en.wikipedia.org/wiki/WebDAV>
- [7] Singh D., Singh J. and Chhabra A., “High Availability of Clouds: Failover Strategies for Cloud Computing Using Integrated Checkpointing Algorithms,” *Communication Systems and Network Technologies (CSNT)*, pp. 698-703, 2012.
- [8] Radojevic B. and Zagar M., “Analysis of Issues with Load Balancing Algorithms in Hosted (Cloud) Environments,” *MIPRO*, pp. 416-420, 2011.
- [9] Bjorkqvist M., Chen L.Y. and Binder W., “Load-Balancing Dynamic Service Binding in Composition Execution Engines,” *Services Computing Conference (APSCC)*, pp. 67-74, 2010.
- [10] Roubos D. and Bhulai S., “Session-Level Load Balancing for High-Dimensional Systems,” *Automatic Control*, pp. 2018-2023, 2009.



