具負載平衡與容錯機制的雲端魯班陽宅評估系統

陳裕升[1]　　高勝助[2]
[1]國立中興大學 資訊科學與工程學系
c9873005@cs.nchu.edu.tw
[2]國立中興大學 資訊科學與工程學系
sjkao@cs.nchu.edu.tw

摘要

本論文在 KVM(Kernel-Based Virtual Machine)虛擬環境下，建構由 Apache 提供負載平衡機制與網路卡 Bonding 技術支援容錯功能的伺服器叢集管理系統。我們以魯班陽宅風水量化評估的應用服務，作為驗證本系統可行性。實作環境由各佈署三部 KVM 虛擬機器之兩台伺服主機，接受前端單一入口的服務分配與資源使用監控主機，提供具負載平衡與容錯機制的魯班陽宅風水量化服務。實驗結果顯示，藉由每 100ms 網路卡的狀態監控，有異常發生時，本系統可以馬上啟動備援機制。而在客戶需求變化時，透過虛擬機器的開關，本系統也可以有效的分攤服務請求，達到負載平衡的效果。

關鍵詞：KVM、負載平衡、容錯機制、雲端運算、魯班陽宅風水

1. 前言

雲端技術之一的虛擬化（Virtualization）廣泛被應用在大量登入系統服務的需求上，這些需求更可預知大量登入系統和負載量造成延宕或無法登入的情況，相反的如購置大量硬體需求的同時確會發生伺服器閒置所造成資源浪費，例如交通運輸的台灣高鐵、台鐵訂票系統、學校單位選課系統、社群網路等短暫尖峰性登入產生的系統瓶頸，如購置軟硬體來因應需求人數，伴隨而來將造成龐大的資本支出，且短暫尖峰過後系統閒置，相對造成資源浪費、電力耗損及設備折舊。

根據[12]雲端網路與虛擬化知名廠商 F5 與 VMware 的解決方案，提供給客戶經由負載平衡設備監控流量的變化開啟新的 VM，再透過 VMware API 介面控制跟負載平衡器註冊新的 VM，將使用者的需求導向新的 VM，此方法在成功案例中，降低了 80-90%的能源成本及能源消耗量。同時，因應雲端頻寬需求大量提升，提供虛擬化與實體網路交換資訊，藉由多片實體網路卡來達到頻寬的負載平衡及容錯功能。兩家廠商所提供的負載平衡與虛擬化整合應用，是論文研究動機的根源。

本論文中藉由前端（Front-end）Apache 核心 mod_proxy[13]模組中的平衡負載機制與網路 bonding 功能，主要為資源分配下，降低主機網路單點失效(Single Point of Failure)；再配合虛擬 KVM(Kernel-Based Virtual Machine)管理機制，應用在大量即時性需求的魯班陽宅量化服務用。探討如何提供魯班風水評估服務，因應大量用戶端連線需求，整合文字界面操控 libvirt 提供自動化調節虛擬機器啟動與關閉；且配合負載平衡將資料分流導向虛擬機器，運作中主要是經由撰寫腳本與 libvirt。

利用雲端虛擬化平台提供魯班風水服務，簡稱魯班雲(Luban Cloud)，是摘錄古本葬經[2]及吳教授開運陽宅與職場風水[3][4]思考出獨特的演算法量化機制;再經由撰寫軟體應用服務，將傳統應用服務平台移轉至雲端應用平台。

本論文提出動態資源調配後端虛擬機器，策略性將雲端虛擬化應用在魯班雲上，當需求增加時提供靈活且自動化啟動，需求減少時則是降低資源浪費。藉由論文提出的架構實作出可降低系統管理員負擔與實體機器資源有效利用的雛形系統。

2. 相關研究

在傳統叢集架構(Clusters)節點主機運算平台大多在單一作業系統，而雲端架構下藉由虛擬化的技術，在單一實體機器上運作不同作業系統與硬體資源共享[8]。雲端架構下建置提供負載平衡的伺服器叢集，以及藉由網路卡容錯的機制，提供魯班風水應用服務，所使用的相關技術如下。

2.1 KVM and QEMU

KVM(Kernel-Based Virtual Machine) 與 QEMU[5]的提出，主要為 Linux 系統上使用的虛擬化技術，運作上如同一般程序(process)，在 User-Space 中的 QEMU 呼叫 ioctl (input/output control) 來控制 I/O 讀寫，並且支援 user application 存取裝置的重要 system call，經由 Linux 核心模組/dev/kvm 與 Guest System 進行溝通。

2.2 Virtio Balloon

針對記憶體資料異動頻繁最佳化設計，目前可靠性的方法有二種，方法一使用一次佔滿多大記憶體，適用於資料庫服務的虛擬機器，但閒置時易造成記憶體閒置的情況。

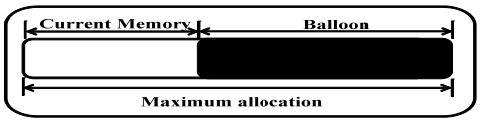


圖 1 Virtio Balloon 記憶體示意圖[9]

方法二是使用 KVM 記憶體核心管理「Virtio Balloon」，會將記憶體分為「Current Allocation」與「Balloon」兩個區塊，如圖 1 所示。前者為系統啟動後記憶體獲得的初始值，後著乃指記憶體的容量可依需求動態調節，因此虛擬系統運作負載提高則 Balloon 容量縮小，反之，若虛擬系統閒置，則記憶體使用量小。2010 年洪婉荏[1]提出記憶體監控機制中，透 SNMP 的方法來動態監控實體與虛擬機器的記憶體負載狀況，就是採用 Virtio Balloon 的方法。

2.3 Libvirt

Libvirt[17]是一在 Linux 作業平台中，提供多種虛擬機器管理的工具，支援多種 hypervisor，包含 KVM/QEMU、XEN、Visual Box、Parallel 等。Libvirt 套件包含 API 套件庫與背景執行程序(Daemon)libvirtd。使用此套件的軟體系統有 virsh、virt-install、virt-viewer、virt-image、virt-manager。延伸性語言開發則可使用 C、Python、Perl、Java 等程式。我們將使用 Virsh 來對虛擬機器進行管理。

2.4 Load Balancing

伺服器主機因應動態的需求，因此需要負載平衡的技術來分散工作執行，以降低回應時間，使客戶端能提供更好的服務品質。知名廠商如 Nortel、Cisco、F5、Foundry、Radware 提供相關硬體支援的方式藉由 L4 到 L7 多層交換器的設備來實現負載平衡。而目前 Linux 上常用負載平衡套件系統有：

- LVS-NAT、Direct Routing、IP Tunneling[14]
- Apache Module (mod_proxy)[13]
- Nginx[15]
- HAProxy[15]

本論文採用 Apache Module 的方式，實作伺服器主機上的負載平衡。

2.5 Bonding

虛擬裝置 Bonding[11]功能最早源於 Linux Kernel 2.0，由 Donald Becker 提供核心修補，後來成為 Linux Kernel 功能一部份；目前主要模式由 0 至 6，共七種模式。而我們主要是透過 Modes 4 模式，遵循 RFC802.3ad[16]，又稱 LACP (Link Aggregation Control Protocol)，主要功用是可倍增頻寬與網路容錯機制。實作建置需選擇有支援協定運作的網路卡及交換器設備。

2.6 風水

郭璞[2]古本葬經內篇提及「氣乘風則散，界水則止。古人聚之使不散，行之使有止，故謂之風水」，是「風水」詞源於最早所被提及，而整段說明位居山脈環抱形成天然的氣場，道理如同房屋四周牆壁要能遮風避雨；但是也需開門或鑿窗，好讓空氣適度流通。論文中藉由吳彰裕教授著書中的吳教授開運陽宅[3]與吳教授職場風水[4]，將魯班陽宅總論說明的居家風水觀點中，我們依此推論風水量化係數，提供雲端環境下，使用者購屋參考的依據，也藉此應用印證所開發系統的可行性。

3.系統架構與模組

本論文利用 Apache 負載平衡機制中 Round Robin 的方式，將請求導向後端虛擬機器的伺服叢集，主要藉由已佈署的虛擬機器來建立服務叢集，虛擬伺服主機中，除固定常駐的機器外，叢集中的機器數量會隨使用者的需求而增減。我們以一台實體主機機器作為單一對外服務窗口並建置網路卡 Bonding 的機制，除了增加頻寬之外，也強化單一網路界面失效的容錯功能。使用者請求服務時，主控端依虛擬機器服務的連線狀況，動態調整虛擬機器數量，避免負載集中在局部虛擬機器上。系統實體架構如圖 2 所示。

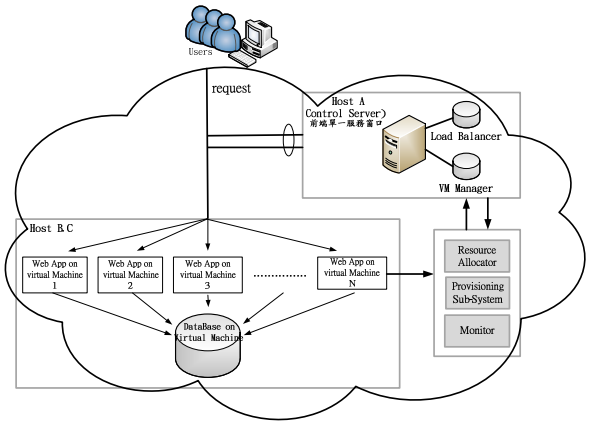


圖 2 系統實體架構圖

3.1 系統功能架構與演算法

雲端虛擬化的動態資源，主要以使用 CPUs、記憶體(Memory)、磁碟容量(Disks)、網路品質(Communication networks)來衡量。此外以目前雲端服務，大多以 WEB 介面的方式來提供服務，根據 WEB 服務的特性，可以如下四種方式來衡量系統資源的使用[6]：

(1)Number of concurrent users.
(2)Number of active connections.
(3)Number of requests per second.
(4)Average response times per request.

本論文採用在單一實體主機上，參考動態資源調節的演算法[7]，針對負載超過可容許虛擬機存活連線數量與閒置狀態下，設計成為一個動態調節虛擬機器。如圖 3 所示。演算法中以最多 6 部虛擬機器，最少 2 部虛擬機器為例。

```
1  For an instance i in Ninstance (Ninstance initialized by 2)
2    If(Ai/SMax>=Tupper) then
3     Increment NExceed
4    If(Ai/SMax>=TLower) then
5     Increment NBelow
6    Record and sort all indexes J in ascending of Ai/SMax
7  If (NExceed== 6) then
8    Send a notification to the admin
9    break;
10   If(NExceed== Ninstance) then
11     Provision and start a new instance
12     Add new instance to load-balancer
13 If(NBelow>=2) then
14     Set m equal first index in j
15     Shutdown instance m
16     Remove instance m form Load-Balancer
17     Decrement number of instances: Ninstance
18
19 For an instance i in Ninstance
20   Evaluate normalized load factor:
21   Apply new load factors Lj to Load Balancer

Where
 Ai:number of active session in instance i
 SMax:Maximum sessions per instance (SMax initialized by 100)
 Tupper:session upper-threshold (Tupper initialized by 60)
 TLower:session lower-threshold (TLower initialized by 10)
 Ninstance:Number of existing instances
 NExceed:Number of instances Exceeding session upper-threshold
 NBelow:number of instances Below session upper-threshold
```

圖 3 演算法-虛擬機器動態資源調整

演算法說明如下：

第 1-6 行中，Nistance 為開啟的總虛擬機器數量預設為 2 台，藉由虛擬機器連線數 Ai 除於虛擬機器最大可連線數 SMax 來計算，排序並進行統計超過上限值的虛擬主機數量 NExceed 或低於下限值 NBelow 的虛擬主機數量，並且進行虛擬主機排序。

第 7-12 行中，如全部虛擬機器連線數都超過上限值，則先判斷全部虛擬機器是否已開啟最大數量 6 台，如果是則送出警告訊息，否則依情境進行開啟虛擬機器，並且將啟動的虛擬機器加入負載平衡叢集(第 19-21)。

第 13-17 行中，如虛擬機器連線數低於下限值，則先判斷單一虛擬主機上的虛擬機器是否有大於等於數量 2 台，如果是則藉由排序索引指向連線數最低的虛擬機器，依情境將虛擬主機關閉，並且將虛擬機器移除負載平衡機制(第 19-21)中，已開啟的虛擬機器統計數量減少一台。

3.2 系統功能模組

本論文藉由主控端監控，提供負載平衡運作機制與觸發啟動關閉後端虛擬主機伺服器的虛擬機器，同時當作對外服務的單一窗口。系統功能模組關係圖如圖 4。

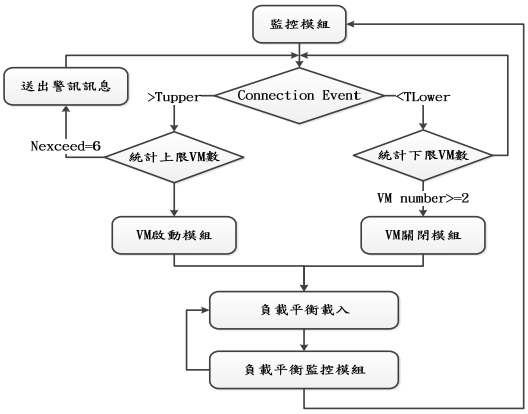


圖 4 系統功能模組關係圖

系統架構流程說明：

步聚一：佈署已建立的虛擬映像檔，虛擬主機上各啟動一台 VM，映像檔容量 8G，配置一顆 vCPU 及記憶體<currentmemory>2G，虛擬機器已安裝軟體套件、腳本排程。

步驟二：監控模組藉由自行撰寫與功能腳本負責監控虛擬主機與虛擬機器運作負載狀況。

步驟三：腳本中設定利用 SSH 的方式取得虛擬機器連線需求數量，每隔五分鐘藉由輪詢（Polling-based）進行虛擬機器連線數量統計與排序資訊，觸發相關模組運作。

步驟四：依定義的需求的門檻值來觸發啟動模組或關閉模組，並且將虛擬機器加入負載平衡模組運作。

步驟五：將虛擬機器的啟動狀況載入至負載平衡模組運作中，事件中如虛擬機器發異常或關閉會持續檢查。

步驟六：本論文提出系統流程運作，包含監控模組、啟動模組、關閉模組及負載平衡模組，並利用腳本週期性監控與執行。

3.3 虛擬機器啟動流程模組

虛擬機器的啟動需要依使用者連線數來進行判斷，目前硬體規格規劃，每台虛擬主機可啟動的虛擬機器數量為三台，兩台虛擬主機最大為六台，因此達到虛擬機器數量上限即發出警告訊息通知。啟動模組運作流程如圖 5。

啟動模組流程說明:

步驟一：預設啟動兩台虛擬主機上會各啟動一台虛擬機器 ，因此總虛擬機器數量為兩台。

步驟二：接著前端會取得虛擬機器上的連線數，兩台虛擬主機上的虛擬機器連線各別進行連線數的統計與排序。

步驟三：如已啟動的虛擬機器上的 WEB 服務連線數都超過定義的 TUpper 的值，則觸發啟動模組。

步驟四：觸發啟動模組首先會計算 VM 數量，如虛擬主機上的 VM 啟動數量相同時，則預設是優先啟動定義 HOST B 上的 VM。

步驟五：虛擬主機的 VM 數量不相同時，則 VM 啟動模組會啟動虛擬主機 VM 數量最少的主機，以利負載均衡分配至虛擬機器。

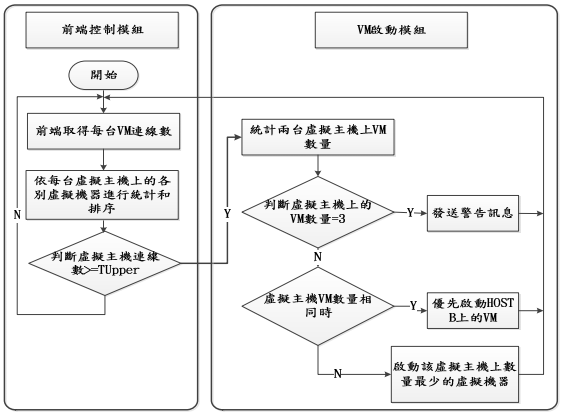


圖 5 啟動模組運作流程

3.3.1 虛擬機器關閉流程模組

虛擬機器的數量需符合良好服務品質要求，如因 WEB 連線數降低而關閉虛擬機器，則需有監控機制來避免誤判，防止兩台虛擬主機低於各啟動一台 VM 的備援需求。關閉模組運作流程如圖 6 所示。

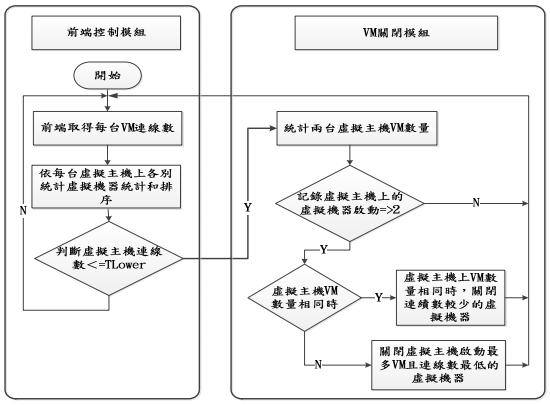


圖 6 關閉模組運作流程

關閉模組流程說明：

步驟一：兩台虛擬主機上各會啟動一台虛擬機器，因此總虛擬機器的啟動數量兩台。

步驟二：主控端主機取得虛擬機器上的連線數，再進行統計和排序，判斷己啟動的虛擬機器連線數是否等於小於 TLower 的值。

步驟三：條件成立則觸發 VM 關閉模組，接著會再次統計虛擬主機上的 VM 數量。

步驟四：判斷虛擬主機上的 VM 啟動數量是否開啟二台或二台以上，當實機主體上的虛擬機器啟動相同時，則經由統計和排序的結果，關閉最低連線數的虛擬機器。

步驟五：如虛擬主機上的 VM 數量不相同時，則會關閉虛擬機器啟動最多且連線數是該虛擬機器 VM 最低。

3.3.2 負載平衡運作流程模組

主控端的核心同時運行負載平衡機制，如要讓新啟動的虛擬機器加入叢集運作，則需更新 Apache 負載平衡，相反的關閉虛擬機器或異常情況，退出叢集運作也需要更新叢集資料庫資訊，才能使得負載平衡與虛擬機器相互配合。負載平衡運作流程如圖 7 所示。

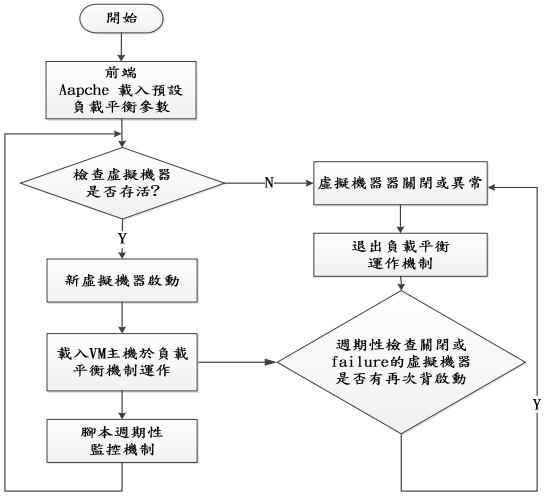


圖 7 負載平衡運作流程

負載平衡運作流程說明：

步驟一：啟動後載入負載平衡輪詢演算法及參數。

步驟二：機制中定期檢查虛擬主機的存活是否正常，如有新的虛擬主機啟動則自動載入至負載平衡機運作中，且持續不斷偵測主機存活狀況。

步驟三：如虛擬主機關閉或異常情況，則退出負載平衡運作機制，但仍然持續偵測 VM 主機是否被啟動的情況，如有 VM 再次被啟動則再次加入 VM 至負載平衡叢集運作。

步驟四：依撰寫腳本配合功能組態，週期性監控虛擬機器運作。

4.實作模擬與測試

4.1 實驗系統架構

系統實作架構如圖 9 所示，由於實作網路容錯機制，因此設備需提供支援 802.3ad 相關協定，主控端 Host A 系統核心使用 Debian Linux server 作業系統，版本為 3.2.0-4-amd64 ，且安裝 Apache 負載平衡套件，Linux 核心 Bonding 功能所需的 ifenslave-2.6 套件，及設定/etc/network/interface 內的 bond0 網路組態，以利將兩張 NIC(Network Interface Card)進行 team 的結合。主機硬體規格如表 1 所示。

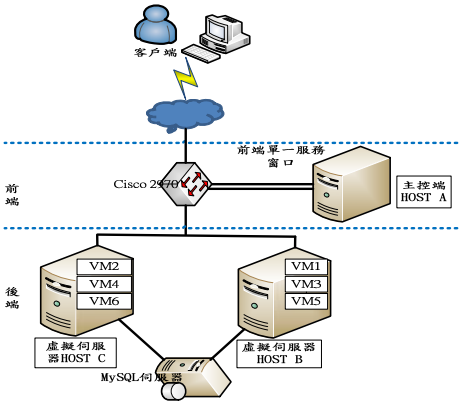


圖 9 系統實作架構

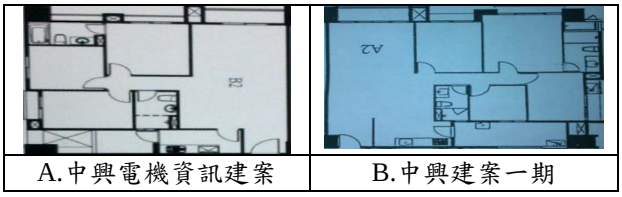
表 1 實體機器規格表

	Host A	Host B、 C
Base Board	Asus P8H67-MPRO/B3	Gigabyte GA-78LMT-USB3
CPU cores	4	6
Processor	Intel Core I5-2400	AMD FX(tm)-6100
Memory	16 G	8 G
Disk	1 TB	1 TB
OS	Debian (wheezy)	

4.2 魯班雲的量化評量

終端使用者的操作說明：

由使用者在操作上需輸入相關房屋資訊，但部份特殊名詞易被遺忘或難於瞭解定義，可應用點選網頁顯示魯班雲定義標準。我們藉由圖 10 (A)(B)位於中興大學校內，輸入相關房屋資訊。



A.中興電機資訊建案　　B.中興建案一期

圖 10 建案實務格局範例圖

圖 11 魯班雲量化分析-外部環境畫面

圖 11 畫面為魯班雲外部環境畫面，包含屋種類型、陰煞、水勢、路勢等，依序輸入目前提供幾項範例指標，之後需提供建築外觀、建築內部資訊。完成後魯班雲系統會進行量化分析運算。

魯班雲係數量化分析：

魯班風水量分析如圖 12 評量分數，兩房都屬於公寓類型且地點模擬於中興校區，因此外部環境大多相同，但建築物內部差異較大。藉由魯班雲提供使用者量化分析數據，購屋置產能提供參考性數據，並能尋求解決之道或擇優而居。

圖 12 魯班雲量化分析-查詢記錄畫面

4.3 網路容錯機制驗證

本論文系統架構由三台實體機器，如圖 9 所示，主控端 Host A 為單一服務窗口，同時負責將使用者需求導向後端虛擬伺服器主機及動態調整虛擬機器的管理，因此需建置主機網路容錯機制，降低因網路硬體故障所造成單點失效（SPOF-Single Point Of Failure），造成服務運作中斷。我們實作兩張網卡 Bonding，並直接交互中斷網路 eth1 或 eth2 之一，驗證網路實際運作的容錯及可靠度，如圖 13 所顯示。

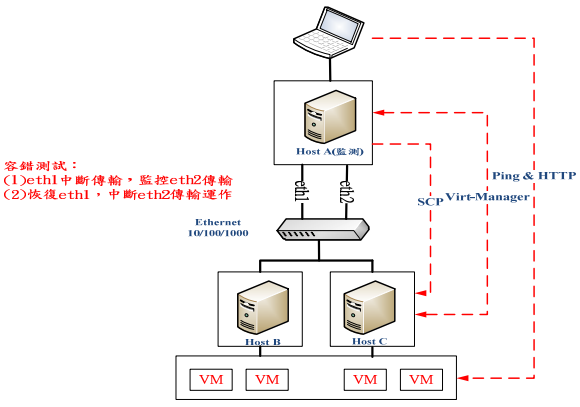


圖 13 中斷網路示意圖

藉由主控端 SCP 傳送一個大檔案至兩台實體主機，並且啟動 virt-Manager 及 NB 主機發送指令 Ping，藉由監測觀察網路容錯情況，其中虛擬主機為 HOST B 與 HOST C，而虛擬伺服器則啟動四台。在此情境下監測兩次手動中斷網路，未發生網路容錯失敗的情況。網路卡 bonding 後的虛擬裝置名稱為 bond0，兩張網卡的封包相加總合為 bond0 封包總量。圖 14 Host A 網路容錯機制驗證。

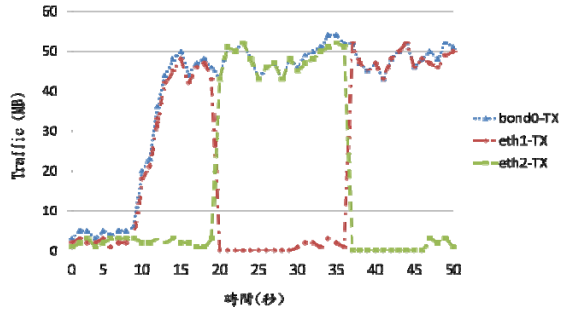


圖 14 Host A 網路容錯機制驗證

4.4 虛擬伺服主機的負載平衡

論文提出的動態虛擬機器調整，進行虛擬伺服器叢集測試，前端主機為獨立的實體主機運作，所有的後端虛擬主機都是相同的硬體規格。而服務請求會由單一入口主機轉遞至後端虛擬機器，叢集主要套用 Round Robin 負載演算法，我們由測試來觀察虛擬機在連線請求下，需求變化而動態增減一台虛擬機，觀察連線需求的負載的情況。

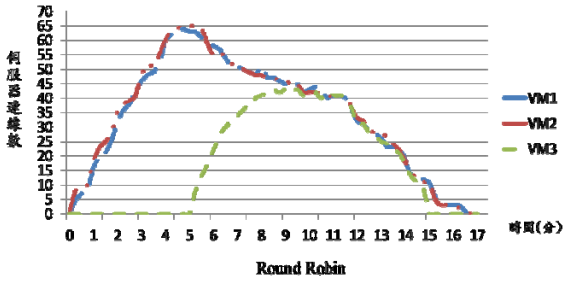


圖 15 連線數請求測試

我們藉由魯班雲的網頁，發出大量連線需求，觸發相關模組機制，預設每台虛擬伺服器主機最大可連線數量為 100，系統測試連線數超過六十即觸發開啟新虛擬機器，低於百分之十則觸發關閉虛擬機器。我們由圖 15 觀察到在第五分鐘時，實際已經超過定義的連線數，因腳本循環機制設定為每五分鐘的監測，實作上，在第五分鐘時啟動新的 VM 後，約五分鐘的時間內，三台 VMs 的連線數可以達到平衡。

當需求降低時，在第十五鐘時發現 VM3 已經低於十條連線的門檻值，因此率先被關閉。之後，VM1 與 VM2 虛擬機器雖低於下限的連線數，但實作環境中會因至少有一部虛擬服務主機在每部主機上的要求，而持續保持開機的狀態。

5. 結論與未來展望

雲端虛擬化與負載平衡機制，可有效利用實體主機資源及樽節成本，並利用網路卡容錯機制達到自動備援，避免造成服務中斷。應用服務部份，目前僅提魯班風水範例評量係數，尚未納入魯班風水全部內容，未來如提供適地性與創新服務，更能增加參考性價值。

雲端服務需考量上網功能的裝置，越來越普及，因此需提供虛擬叢集更加彈性的服務，我們藉由魯班雲服務平台連線數來動態調整虛擬機，主要由論文提出的演算法在適宜時機啟動關閉虛擬機，並套用 Round Robin 機制作負載平衡，可以有效分攤服務請求。未來研究加入更多種負載平衡排程演算法比較其效益，並套用不同 Bonding 容錯機制，如 Balance-rr 或 Balance-xor，提供評估可靠性服務與最佳化動態資源調整。

參考文獻

[1] 洪婉荏，"一個具自動化部置與動態調節資源的虛擬機器管理平台"，國立中興大學資訊科學與工程研究所，碩士論文，2011。

[2] 郭璞，地理善本葬經，大山書局，1999。

[3] 吳彰裕，吳教授開運陽宅，時報出版，2005。

[4] 吳彰裕，吳教授開運職場風水，時報出版，2006。

[5] Goto, Yasunori, "Kernel-based Virtual Machine Technology," Fujitsu Sci. Tech. J, 47.3,pp.362 -368, 2011.

[6] Chieu, T. C, Mohindra, A, Karve, A. A., & Segal, A, "Dynamic scaling of web applications in a virtualized cloud computing environment," IEEE International Conference on e-Business Engineering (ICEBE), 2009.

[7] Chieu, Trieu C, Ajay Mohindra, and Alexei A. Karve, "Scalability and Performance of Web Applications in a Compute Cloud," IEEE 8th International Conference on e-Business Engineering(ICEBE), 2011.

[8] Buyya, R, Yeo, C. S, Venugopal, S., Broberg, J., & Brandic, I, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," Future Generation computer systems, 25(6), pp.599-616, 2009.

[9] Linux KVM-記憶體最佳化管理與應用, available from: http://linuxkvm.blogspot.tw/2011/06/linux-kvm_19.html, June 2011.

[10] Configuring IEEE 802.3ad Link Bundling and Load Balancing, available from: http://www.cisco.com/en/US/docs/ios/cether/configuration/guide/ce_lnkbndl.html, June 2012.

[11] Linux Ethernet Bonding, available from: http://www.kerne-l.org/doc/Documentation/networking/bonding.txt, April 2011.

[12] F5 Networks, available from: http://www.f5.com.tw/pdf/white-papers/f5-vmware-green-it.pdf, May 2008.

[13] Apache Module mod_proxy , available from: http://httpd.apache.org/docs/2.2/mod/mod_proxy.html#proxypass, 2013.

[14] Linux Virtual Server (LVS),available from: http://www.linuxvirtualserver.org/index.html, Aug. 2012.

[15] Comparing Nginx and HAProxy for web applications, available from: http://affectioncode.wordpress.com/2008/06/11/comparing-nginx-and-haproxy-for-web-applications/, June 2008.

[16] IEEE 802.3 ETHERNET WORKING GROUP, available from: http://www.ieee802.org/3/index.html, Mar. 2013.

[17] Libvirt virtualization API, available from: http://libvirt.Org/index.html, 2013.