伺服器叢集：技術與系統

張智晴　林盈達

國立交通大學資訊科學研究所

摘要

伺服器叢集是一群伺服器的集合，有兩個主要功能，負載平衡與容錯，負載平衡可以把使用者的要求分散給伺服器叢集中的伺服器處理，使得網站可以接受較多的連線；容錯可以是伺服器容錯或是檔案系統的容錯，伺服器容錯的技術是指當主要伺服器當掉或無法繼續提供服務時，備份伺服器會啓動並取代主要伺服器，使得服務不中斷。而檔案系統的容錯是指當檔案系統發生損毀時，可以經由記錄或其它方法回復。

在這篇報告中，我們觀察了目前現有跟叢集技術有關的系統，首先會介紹叢集技術及實做叢集技術的方法，接著我們會比較現有的各種伺服器叢集系統，然後介紹其中較具代表性的幾種，並詳細介紹 Piranha 這個伺服器叢集系統，包括其安裝流程，最後做出結論。

關鍵字：可靠度、伺服器、伺服器叢集、負載平衡、容錯、DNS、HTTP Redirect、NAT、IP Tunneling、Direct Routing 、Requests Proxy 。

1. 簡介

隨著網際網路(Internet)的快速發展，連接網際網路的主機數目與參與網際網路的人數皆與日俱增，而架構在網際網路上的全球資訊網(WWW, Web Wild Web)，其使用量更是呈現爆炸性成長。在這個電子商務(E-Commerce)盛行的時代，一個全球資訊網網站所提供的服務愈來愈多，並且必須面對全世界使用者的連線，可是網站的可靠度(reliability)卻沒有跟上使用者成長的速度，常常因爲系統無法處理過多的要求(request)而當掉或是無法繼續提供服務。

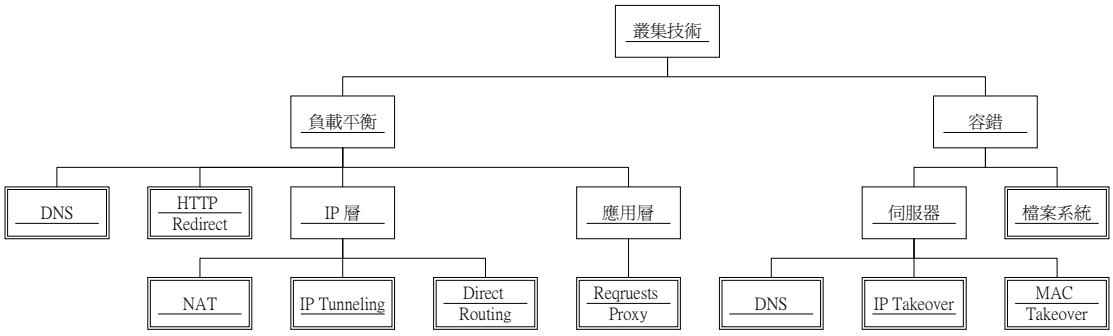
爲了提高網站的可靠度，有許多方法被提出來，伺服器叢集(server clustering)技術是其中一

種。伺服器叢集是一群伺服器的集合，有兩個主要功能，負載平衡(load balancing)與容錯(fault tolerance)，負載平衡可以把使用者的要求分散給伺服器叢集中的伺服器處理，使得網站可以接受較多的連線；容錯可以是伺服器容錯或是檔案系統的容錯，伺服器容錯的技術是指當主要(primary)伺服器當掉或無法繼續提供服務時，備份(backup)伺服器會啓動並取代主要伺服器，使得服務不中斷。而檔案系統的容錯是指當檔案系統發生損毀時，可以經由記錄(log)或其它方法回復。

在這篇報告中，我們觀察了目前現有跟叢集技術有關的系統，首先會介紹叢集技術及實做叢集技術的方法，接著我們會比較現有的各種伺服器叢集系統，然後介紹其中較具代表性的幾種，並詳細介紹 Piranha [1]這個伺服器叢集系統，包括其安裝流程，最後做出結論。

2. 叢集技術：負載平衡及容錯

負載平衡與容錯是叢集技術的兩個主要功能，可以增加系統的可靠度，在這一節中，我們整理了目前各種實做叢集技術的方法，如圖一所示。



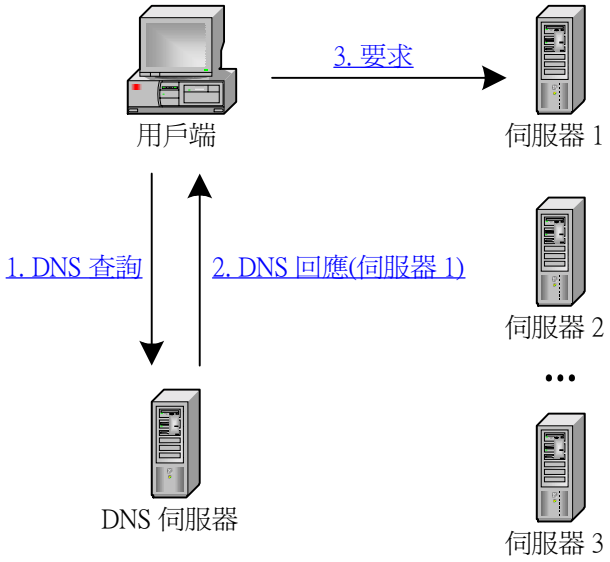
圖一、叢集技術的種類

我們將實做叢集技術的方法分成兩類，負載平衡與容錯。實做負載平衡的方法有 DNS (Domain Name Service)、HTTP (HyperText Transmission Protocol) Redirect [2]、NAT(Network Address Translation)、IP Tunneling、Direct Routing 與 Requests Proxy。其中，NAT、IP Tunneling 和 Direct Routing 屬於 IP 層(IP Level)的負載平衡，而 Requests Proxy 則屬於應用層(Application Level)的負載平衡。IP 層的負載平衡是指負載平衡的程度做到 IP 層，可以做到每個連線(per connection)的負載平衡，而應用層的負載平衡程度比較細，可以做到每個要求(per request)的負載平衡。因此應用層負載平衡的負載平衡效果會比較好，但是相對的，它必須付出較大的額外成本(overhead)。

實做容錯的方法有 DNS、IP Takeover、MAC Takeover 與檔案系統(File System)。其中，DNS、IP Takeover 和 MAC Takeover 是屬於伺服器的容錯。

2.1 負載平衡：使用 DNS

這類型的負載平衡是利用 DNS 對映網域名稱(Domain Name)到 IP 位址的機制來做到負載平衡。圖二可以說明它的運作方式，當用戶端(client)以網域名稱要求連線時，必須先向 DNS 伺服器查詢 IP 位址，接著 DNS 伺服器就根據它的排程演算法(scheduling algorithm)把伺服器叢集中某一台伺服器的 IP 位址回傳給用戶端，用戶端再向這個 IP 位址的伺服器要求連線。



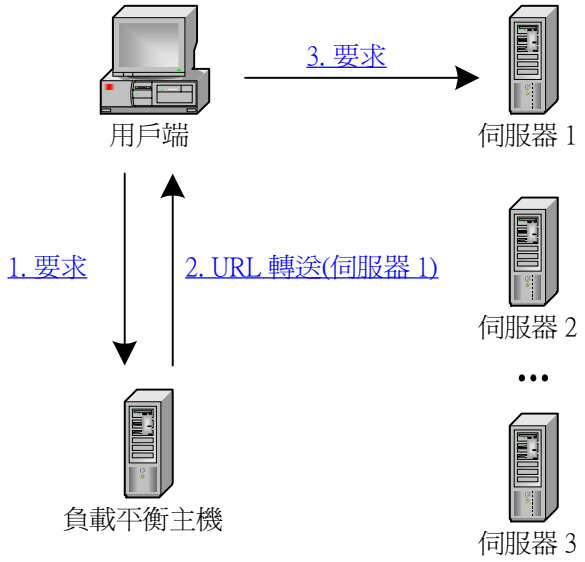
圖二、負載平衡：使用 DNS

使用這個方式來實做負載平衡相當容易，但卻有負載不平衡的問題。DNS 架構是階層式的(hierarchical)，因此當一用戶端使用某一遠端網站的服務時，用戶端會先向本地(local) DNS 伺服器查詢該遠端網站的 IP 地址，如果本地 DNS 伺服器找不到該遠端網域名稱，本地 DNS 伺服器會向更上層的 DNS 伺服器查詢，最後由該遠端網站的 DNS 伺服器選擇其伺服器叢集中某一伺服器之 IP 位址回覆。爲了改善效率，DNS 伺服器使用快取(caching)機制，也就是將之前查詢過的網域名稱和相對映的 IP 位址記錄下來，當下次有相同網域名稱的查詢時，就可以直接回應 IP 位址而不須向更上層查詢。因此，由於 DNS 的階層式架構和快取機制，本地 DNS 伺服器在短時間內只會回應相同的 IP 位址，造成遠端網站的 DNS 伺服器的負載平衡努力失效。這個問題可以經由調整 DNS 項目(entry)中的存活時間(TTL, Time-To-Live)值來解決，把 TTL 的值設小一點可以

減低快取機制造成的影響，但是又引發另一個問題，加大本地 DNS 伺服器的負載(load)。

2.2 負載平衡：使用 HTTP Redirect

這類型的負載平衡是利用 HTTP Redirect 的機制來做到負載平衡。圖三可以說明它的運作方式，當用戶端向負載平衡主機(Load Balancer)要求連線時，負載平衡主機會回給用戶端一個 URL (Uniform Resource Locator)轉送(redirect)指令，而轉送的目的地主機是由負載平衡主機根據它的排程演算法在伺服器叢集中選出的，接著用戶端再向這個伺服器要求連線。



圖三、負載平衡：使用 HTTP Redirect

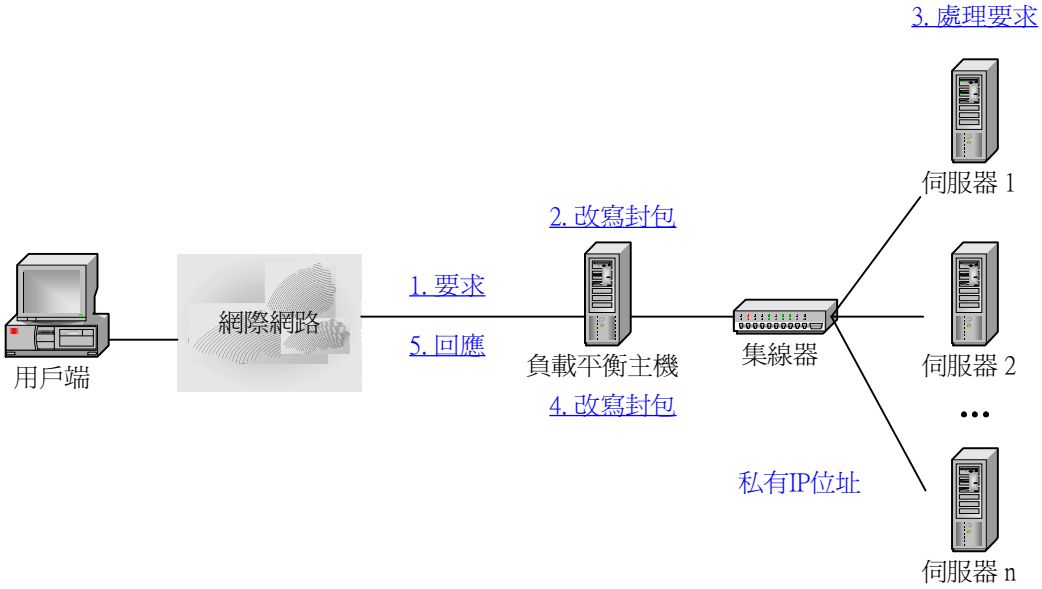
使用這個方式來實做負載平衡也相當容易，不過能做到的負載平衡程度比較低，只能做到每個 session[1]的負載平衡。

2.3 負載平衡：使用 NAT

這類型的負載平衡是利用 NAT，也就是改寫 IP 位址來做到負載平衡。圖四可以說明它的運作方式，在這個架構中，負載平衡主機須要兩個網路介面，一個使用真實(public) IP 位址連接 Internet，一個使用私有(private) IP 位址連接同樣使用私有 IP 的伺服器叢集。當用戶端要求連線時，負載平衡主機會根據它的排程演算法改寫封包，把目地(destination) IP 位址改寫成伺服器叢

[1] 這裡的 session 是指使用者瀏覽一個網站從開始到離開的過程

集中某一台伺服器的 IP 位址，然後把這個封包轉送(forward)給選出的伺服器，並把這個連線記錄到一個用來記錄已建立連線的雜湊表(hash table)，根據這個雜湊表，負載平衡主機才能把接下來的封包轉送給先前被指定處理各個連線的伺服器，伺服器處理完要求後，會將回應送回給負載平衡主機，接著負載平衡主機再把封包的來源 IP 位址改寫成自己的 IP 位址送回用戶端。最後連線結束後，這個連線的紀錄會從雜湊表被移除。



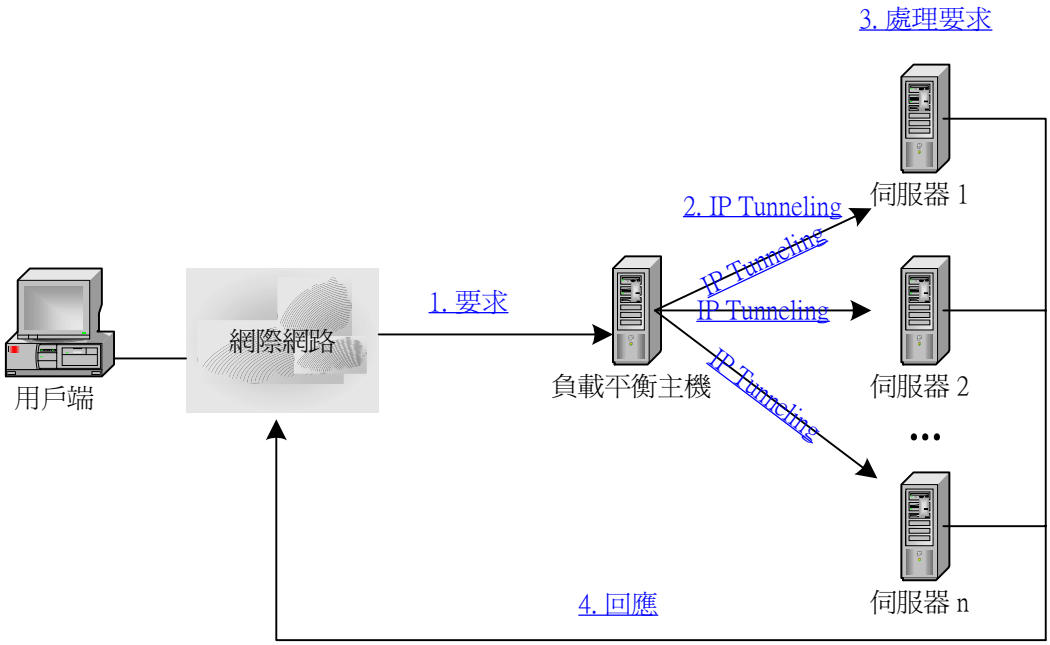
圖四、負載平衡：使用 NAT

使用 NAT 來實做負載平衡是相當方便的方法，因爲伺服器叢集中的伺服器不需要特別的限制，而且由於使用私有 IP 位址，可以解決 IP 位址不足的問題及較具安全性。但是由於封包的分配及回應都必須經過負載平衡主機，所以負載平衡主機將變成瓶頸(bottleneck)，造成伺服器叢集中的伺服器數量的限制。

2.4 負載平衡：使用 IP Tunneling

這類型的負載平衡是利用 IP Tunneling 的技術來做到負載平衡。所謂的 IP Tunneling 技術是指將封包再加上一層 IP 標頭(header)，達到封包轉送的功能。圖五可以說明它的運作方式，在這個架構中，負載平衡主機只須要一個網路介面，但它與伺服器叢集中的伺服器都要各自使用真實 IP 位址及使用 IP 別名(IP alias)共享一個真實 IP 位址，並且需要有 IP Tunneling 能力。當用戶端要求連線時，負載平衡主機會根據它的排程演算法，以伺服器叢集中某一台伺服器的 IP 位址爲

目的 IP 位址把封包封裝(encapsulate)起來，然後把這個封包轉送給選出的伺服器，並把這個連線記錄到一個用來記錄已建立連線的雜湊表，根據這個雜湊表，負載平衡主機會把接下來的封包轉送給處理各個連線的伺服器，伺服器處接到封包後，必須先解封裝(decapsulate)再處理，然後會直接把結果送回給用戶端，最後連線結束後，這個連線紀錄會從雜湊表被移除。



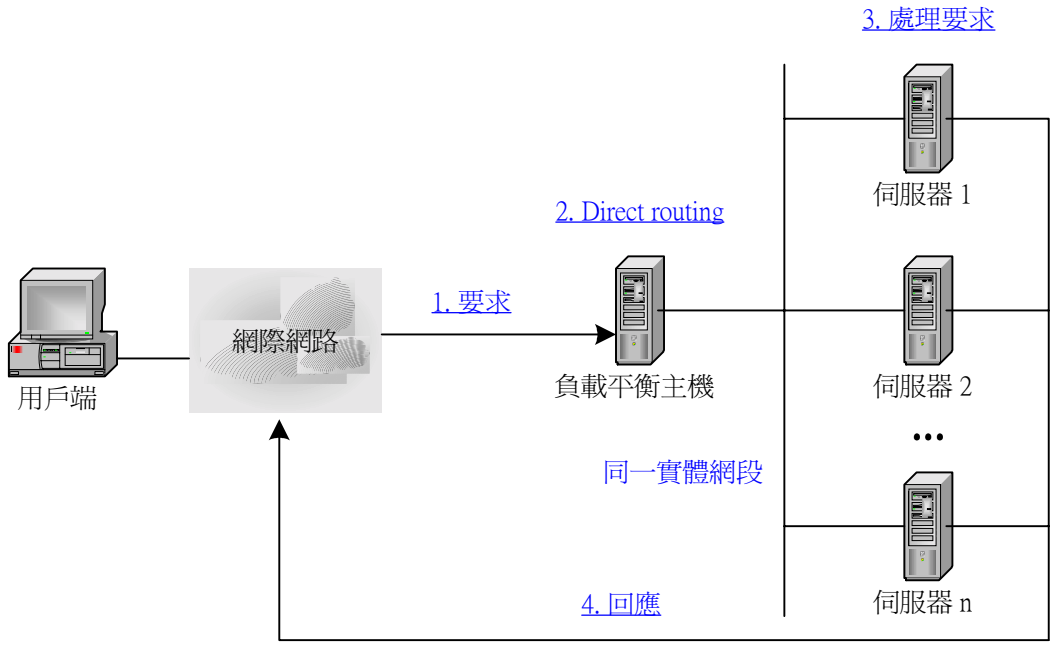
圖五、負載平衡：使用 IP Tunneling

使用 IP Tunneling 來實做負載平衡效能(performance)會比使用 NAT 來得好，因爲伺服器叢集中的伺服器處理完要求後，會直接把結果送回給用戶端，而不需在經過負載平衡主機。但是這個方式有二個限制，第一個限制是伺服器叢集中的伺服器必須使用真實 IP 位址，第二個限制是負載平衡主機與伺服器叢集中的伺服器都需具備 IP Tunneling 的能力。

2.5 負載平衡：使用 Direct Routing

這類型的負載平衡是利用 Direct Routing 的技術來做到負載平衡。這裡的 Direct Routing 是指改寫框架(frame)的 MAC 位址，達到封包轉送的功能。圖六可以說明它的運作方式，在這個架構中，負載平衡主機只須要一個網路介面，但它與伺服器叢集中的伺服器都要各自使用真實 IP 位址及使用 IP 別名共享一個真實 IP 位址，並且要在同一個實體網段(physical segment)中。當用戶端要求連線時，負載平衡主機會根據它的排程演算法改寫框架，把目地 MAC 位址改寫成伺服器

叢集中某一台伺服器的 MAC 位址，然後把這個封包轉送給選出的伺服器，並把這個連線記錄到一個用來記錄已建立連線的雜湊表，根據這個雜湊表，負載平衡主機會把接下來的封包轉送給屬於各個連線的伺服器，伺服器處理完要求後，直接把結果送回給用戶端，最後連線結束後，這個連線會從雜湊表被移除。



圖六、負載平衡：使用 Direct Routing

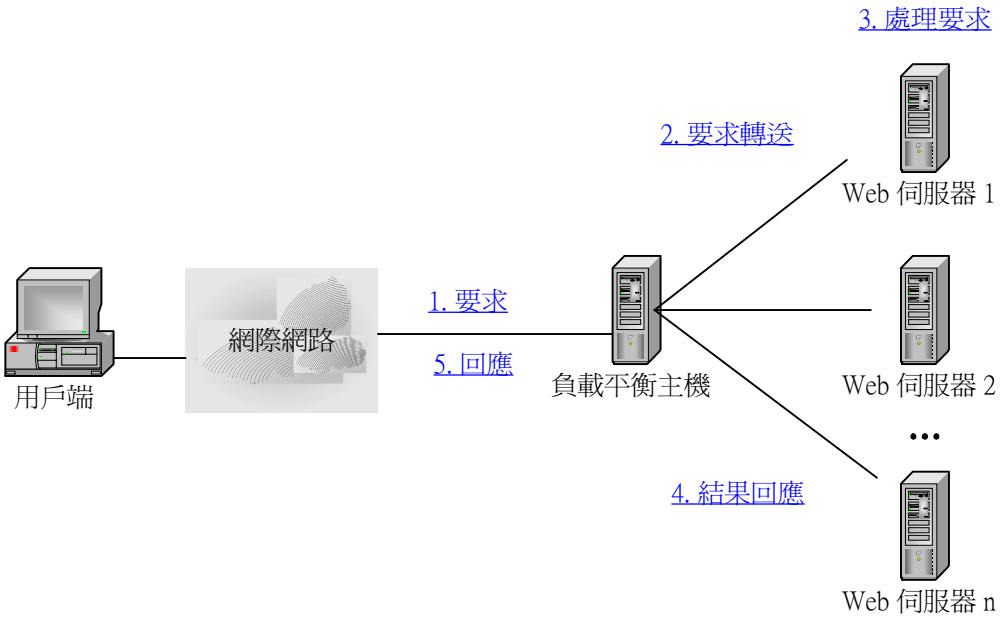
使用 Direct Routing 來實做負載平衡效能又會比使用 IP Tunneling 好一點，雖然這兩個方式在伺服器叢集中的伺服器處理完要求後，都會直接把結果送回給用戶端，不需在經過負載平衡主機，但是使用 Direct Routing 只需要改寫 MAC 位址，而使用 IP Tunneling 必須要經過 IP 封裝/解封裝的程序，所以使用 Direct Routing 效能會比較好。另外，使用 Direct Routing 這個方式也是有二個限制，第一個限制是負載平衡主機與伺服器叢集中的伺服器都要使用真實 IP 位址，並且要在同一個實體網段中。第二個限制是由於負載平衡主機與伺服器叢集中的伺服器都使用 IP 別名共享一個真實 IP 位址而且都在同一個實體網段，所以伺服器叢集中伺服器不能回應這個共享真實 IP 位址的 ARP(Address Resolution Protocol)查詢要求[2]，只能由負載平衡主機回應，否則負載平衡將無法運作。我們把這種不回應 ARP 查詢要求的裝置(device)稱爲 non-arp 裝置。另外，如

[2] 由 IP 位址查詢 MAC 位址

果使用 IP Tunneling 而伺服器叢集中的伺服器也和負載平衡主機在同一個實體網段，也必須要有這個限制。

2.6 負載平衡：使用 Requests Proxy

這類型的負載平衡是利用 Requests Proxy 的技術來做到負載平衡。圖七可以說明它的運作方式，在這個架構中，負載平衡主機只須要一個網路介面，但它與伺服器叢集中的伺服器都要使用真實 IP 位址。當用戶端要求連線時，負載平衡主機會與用戶端建立連線，然後根據它的排程演算法把用戶端的要求經由事先或臨時與伺服器叢集中某一台伺服器建立的連線轉送給它處理，伺服器處理完要求後，會將結果送回給負載平衡主機，接著在由負載平衡主機把結果送回給用戶端。



圖七、負載平衡：使用 Requests Proxy

使用 Requests Proxy 來實做負載平衡可以做到較好的負衡平衡，因爲它的負載平衡是每個要求的負載平衡，而不是每個連線或每個 session (per session)。但是它的效能會比較差，因爲要求與回應都要經過負載平衡主機。另外，使用這個方式有一個限制，必須在伺服器叢集中的伺服器安裝特定的軟體，才能與負載平衡主機建立連線。

2.7 負載平衡：比較

表一是之前所介紹六種實做負載平衡方法的比較表。第一列的層是指負載平衡的程度，第二列是伺服器叢集中伺服器的限制，第三列是伺服器叢集網路限制，第四列是伺服器叢集的個數，第五列是伺服器回應是否要經過負載平衡主機，最後是負載平衡的粗細。

	DNS	HTTP Redirect	NAT	IP Tunneling	Direct Routing	Requests Proxy
層	--	--	IP	IP	IP	應用
伺服器限制	無	無	無	IP Tunneling	Non-arp 裝置	安裝軟體
網路限制	無	無	私有 IP 位址	區域網路	區域網路	無
伺服器個數	多	多	少	多	多	少
伺服器回應	否	否	是	否	否	是
負載平衡粗細	每個 session	每個 session	每個連線	每個連線	每個連線	每個要求

表一、六種實做負載平衡方法較表

2.8 容錯

容錯可以分成兩類，伺服器容錯與檔案系統容錯。其中，伺服器容錯有三種實做方法，使用 DNS、IP Takeover 和 MAC Takeover。使用 DNS 來實做伺服器容錯相當類似之前所提到的使用 DNS 來實做負載平衡，DNS 伺服器會監控主要伺服器，當用戶端向 DNS 伺服器查詢 IP 位址時，如果主要伺服器正常，就回應給用戶端主要伺服器的 IP 位址，反之，如果主要伺服器當掉或無法連線，DNS 伺服器就回傳給用戶端備份伺服器的 IP 位址。第二個方式是使用 IP Takeover 來實做伺服器容錯，主要伺服器和備份伺服器共用一個 IP 位址，備份伺服器會監控主要伺服器，正常狀態下，由主要伺服器提供服務，當主要伺服器當掉或無法連線時，備份伺服器就會取代主要伺服器繼續提供服務。第三種方式是使用 MAC Takeover 來實做伺服器容錯，這個做法跟使用 IP Takeover 很類似，差別只在於一個是共用 IP 位址而另一個是共用 MAC 位址。目前最常用的是 IP Takeover。

檔案系統的容錯是指特別設計的檔案系統，具有較佳的容錯能力，能在檔案系統發生損毀時，經由記錄或其它方法回復，通常也具有處理伺服器叢集中資料一致(consistency)和同步(synchronization)問題的能力。

3. 現有系統分析比較

表二是現有伺服器叢集系統的分析比較表。其中，只要是有關伺服器叢集的技術、軟體，我們都列入比較，不管是整合性軟體或是單一功能套件。比較項目包括，軟體或硬體、商業產品或開放程式碼軟體、負載平衡方式、容錯方式和相關網頁。

	軟體或硬體	商業產品或開放程式碼軟體	負載平衡方式	容錯方式	相關網頁
Linux Virtual Server Project	軟體	開放程式碼軟體	NAT、IP Tunneling 與 Direct Routing	--	http://www.linuxvirtualserver.org/
High-Availability Linux Project	軟體	開放程式碼軟體	--	IP Takeover	http://linux-ha.org/
Red Hat High Availability Server Project	軟體	開放程式碼軟體	NAT、IP Tunneling 與 Direct Routing	IP Takeover	http://people.redhat.com/kbarrett/HA/
Ultra Monkey	軟體	開放程式碼軟體	NAT、IP Tunneling 與 Direct Routing	IP Takeover	http://ultramonkey.sourceforge.net/
FAKE	軟體	開放程式碼軟體	--	IP Takeover	http://www.au.vergenet.net/linux/fake/
The Eddie Mission	軟體	開放程式碼軟體	Requests Proxy	IP Takeover	http://www.eddieware.org/
Coda File System	軟體	開放程式碼軟體	--	File System	http://www.coda.cs.cmu.edu/
InterMezzo	軟體	開放程式碼軟體	--	File System	http://inter-mezzo.org/
Global File System	軟體	開放程式碼軟體	--	File System	http://www.sistina.com/gfs/
LinLogFS	軟體	開放程式碼軟體	--	File System	http://www.complang.tuwien.ac.at/czezatke/lfs.html
TurboCluster Server	軟體	開放程式碼軟體	IP Tunneling 與 Direct Routing	IP Takeover	http://community.turbolinux.com/cluster/
A Scalable HTTP Server: The NCSA Prototype	軟體	開放程式碼軟體	DNS	File System	http://www.ncsa.uiuc.edu/InformationServers/Conferences/CERNwww94/www94.ncsa.html
The Backhand Project	軟體	開放程式碼軟體	Requests Proxy	--	http://www.backhand.org/
IBM Network Dispatcher	軟體	商業產品	Direct Routing	--	http://www-4.ibm.com/software/network/dispatcher/
Cisco DistributedDirector	硬體	商業產品	NAT	--	http://www.cisco.com/univercd/cc/td/doc/pcat/dd.htm
Cisco LocalDirector	硬體	商業產品	DNS 與 HTTP Redirect	--	http://www.cisco.com/univercd/cc/td/doc/pcat/ld.htm
Alteon ACEdirector Web Switch	硬體	商業產品	NAT	--	http://www.alteonwebsystems.com/products/acedirector/
F5 BIG-IP	硬體	商業產品	NAT	IP Takeover	http://www.bigip.com/
Polyserve Understudy	軟體	商業產品	DNS	IP Takeover	http://www.polyserve.com/prod_overview.html
High-Availability.Com	軟體	商業產品	--	IP Takeover	http://www.rsi.co.uk/products/rsf/rsf-linux.html
IBM WebSphere Performance Pack	軟體	商業產品	Direct Routing	File System	http://www-4.ibm.com/software/webservers/perfpack/about.html
Resonate Central Dispatch	軟體	商業產品	NAT、IP Tunneling 與	IP Takeover	http://www.resonate.com/products/central_dispatch/

			Direct Routing		
Mod_Redundancy	軟體	商業產品	--	IP Takeover	http://www.ask-the-guru.de/
Legato Cluster	軟體	商業產品	NAT	IP Takeover	http://www.legato.com/products/availability/legatocluster/
ArrowPoint Content Smart? Web Switching	硬體	商業產品	Requests Proxy	--	http://www.arrowpoint.com/products/switches/index.html

表二、現有伺服器叢集系統分析比較表

4. 八種現有系統的介紹

這一節我們將會介紹八種代表性的伺服器叢集系統。包括 The NCSA Prototype [3]、Linux Virtual Server Project [4]、Cisco's LocalDirector and DistributedDirector [5]、The Eddie Mission [6]、The Backhand Project [7]、High-Availability Linux Project [8]、Ultra Monkey [9]與 Red Hat High Availability Server Project。這些系統有商業產品，也有開放程式碼軟體，有整合性軟體，也有單一功能套件。

4.1 The NCSA Prototype

這是一個相當早期的負載平衡系統，它使用了 Round-Robin DNS 來實做負載平衡和使用 AFS (Andrew File System)這個分散式檔案系統(DFS, Distributed File System)來負責檔案系統容錯和處理資料一致性的問題。它的 Round-Robin DNS 就是之前介紹過的使用 DNS 來實做負載平衡，只是它的排程演算法是使用 Round-Robin，也就是讓伺服器叢集中的伺服器一個接一個輪流服務使用者。較新的 DNS 伺服器都已經支援這個機制。

4.2 Linux Virtual Server Project

這是一個架構在 Linux 作業系統和伺服器叢集系統上的計劃，目標是提供良好的擴充性(scalability)、可靠度和可服務能力(serviceability)。這個計劃所完成的套件皆在 GPL (GNU General Public License)規範下發行，也就是不可進行 private licensing 的開放程式碼軟體，目前已完成軟體有 ipvsadm 這個負載平衡套件。這個軟體包含了三種實做負載平衡技術，使用 NAT、使用 IP Tunneling 和使用 Direct Routing。另外，它提供了四種排程演算法，Round-Robin、Weighted Round-Robin、Least-Connection 和 Weighted Least-Connection。Round-Robin 我們介紹過了，而

Weighted Round-Robin 的差別在於可以設定輪流的順序，Least-Connection 是指挑選伺服器叢集中目前接受最少連線的伺服器來服務使用者，而 Weighted Least-Connection 則是指每個伺服器可以被設定一個權重(weight)值，然後挑選已連線數除以這個權重值結果最小的來接受新的連線。

4.3 Cisco's LocalDirector and DistributedDirector

LocalDirector 和 DistributedDirectior 是 Cisco 兩個用來做負載平衡的產品。屬於商業軟體，並和硬體一起出售。LocalDirector 是使用 NAT 來實做負載平衡，它的排程演算法是根據回應時間，它會測試與伺服器叢集中伺服器的回應時間，然後選回應時間最小的來接受連線。DistributedDirectior 有兩模式可以選擇，DNS Caching Name-Server Mode 和 HTTP Session Redirector Mode，當它操作在 DNS Caching Name-Server Mode 下時，它是使用 DNS 來做負載平衡，排程演算法是根據用戶端位址，它會從伺服器叢集中選出最靠近用戶端的伺服器，然後把 IP 地址回傳給用戶端；當它操作在 HTTP Session Redirector Mode 下時，它是使用 HTTP Redirect 來做負載平衡，而排程演算法和 DNS Caching Name-Server Mode 相同。

4.4 The Eddie Mission

這是一個由易利信(Ericsson)贊助的伺服器叢集計劃，提供百分之百的軟體解決方案。這個計劃的軟體都是由一個函數式(functional)程式語言 Erlang 寫成的，並開放程式原始碼。它包含兩個套件，The Intelligent HTTP Gateway 與 The Enhanced DNS Server。The Intelligent HTTP Gateway 是使用 Requests Proxy 來實做負載平衡，排程演算法是根據伺服器負載，伺服器叢集中的伺服器會定期把自己的負載資訊傳給負載平衡主機，然後負載平衡主機再選出負載最小的來處理要求。另外，它可以靠 admission control 來做到一部分的 QoS (Quality of Service)。The Enhanced DNS Server 是使用 DNS 來做負載平衡，而排程演算法和 The Intelligent HTTP Gateway 相同。

4.5 The Backhand Project

這個計劃也是一個百分之百的軟體解決方案，目前完成的軟體是以 C++寫成的一個 Apache Web Server 模組(module) mod_backhand，並且開放原始程式碼。mod_backhand 是使用 Requests

Proxy 來實做負載平衡，它的排程作法是以一個陣列(array)來代表伺服器叢集中的伺服器，在每次負載平衡時就把這個陣列送到事前指定的排程演算法，它稱爲 Candidacy Function，這些 Candidacy Function 會根據自己的演算法將這個陣列排序、減小或加大，最後取陣列第一個元素所代表的伺服器來處理使用者要求。

它有六種內建的 Candidacy Function，分別是 byAge、byRandom、byLogWindow、byCPU、byLoad 與 byCost，而且可以自定加入新的 Candidacy Function。在這個架構中，伺服器叢集中的伺服器會定期向負載平衡主機報告自己的資訊，包括 CPU 使用率、系統負載和系統記憶體使用量等類似資訊。其中，byAge 可以讓你指定一個值，然後把伺服器陣列中回報資訊時間間隔大於這個值的伺服器刪除，byRandom 是將伺服器陣列亂數排序一次，byLogWindow 是把伺服器陣列的大小取對數，然後刪除掉多的元素，byCPU 把伺服器陣列中 CPU Idle 時間大於某個指定數值的伺服器刪除，byLoad 與 byCPU 類似，它是根據系統負載，最後，byCost 會把要求送給伺服器陣列中所需成本(cost)最低的伺服器。

mod_backhand 有一個特別的優點，它是屬於 Multi-point 負載平衡，也就是伺服器叢集中的每一台都是負載平衡主機。這個特點可以做到更高程度的負載平衡，也可以有相當程度的伺服器容錯，避免單一負載平衡主機會產生的 Single-point failure[3]。

4.6 High-Availability Linux Project

這是一個架構在 Linux 作業系統和伺服器叢集系統上的計劃，目標是提供良好的可靠度、可用性(availability)與可服務能力。目前已完成軟體有 heartbeat 這個伺服器容錯套件，它也是屬於 GPL 軟體。heartbeat 可以透過 PPP 或 UDP 來監控負載平衡主機，並使用 IP Takeover 的方式來實做伺服器容錯。

4.7 Ultra Monkey

Ultra Monkey 是 VA Linux 的一個產品(product)，同時也是一個使用 Linux 作業系統上開放程

[3] 伺服器叢集系統中的某一個元件無法運做就造成整個系統無法繼續服務，通常是指負載平衡主機

式原始碼套件來建立伺服器叢集的計劃，所以它也是開放程式原始碼。這是一個整合性套件，它的架構包括，使用之前提到的 Linux Virtual Server 來提供負載平衡，和 High-Availability Linux Project 的 heartbeat 來提供伺服器容錯。

4.8 Red Hat High Availability Server Project

這是一個整合 RedHat Linux、Piranha 伺服器叢集系統和 Linux Virtual Server(LVS)軟體的計劃，提供高度可用性(HA, High Availability)服務。Piranha 是這個計劃所完成的伺服器叢集系統，它是 RedHat Linux 的一個產品，同時也是開放程式原始碼。這也是一個整合性套件，特點是它提供了一個 Web 界面圖型化系統方便系統管理者設定伺服器叢集系統。我們在下一節會詳細介紹它的組成元件和安裝。

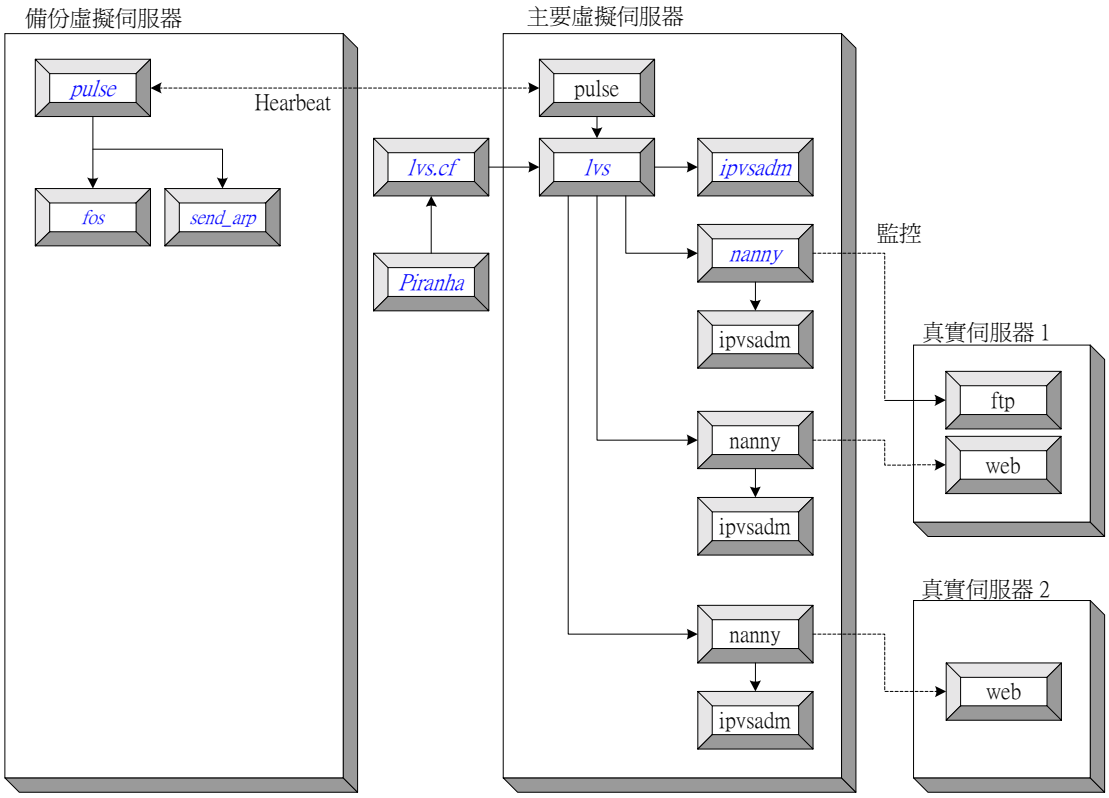
5. Piranha 的介紹與安裝

在這一節中，我們會介紹 Piranha 的組成元件和安裝步驟。圖八是 Piranha 的組成元件與彼此關係架構圖，斜體字部分即是 Piranha 的組成元件，包括 fos、pulse、send_arp、lvs.cf、Piranha、lvs、ipvsadm 與 nanny。另外，Piranha 將伺服器叢集定義成二個部分，虛擬伺服器(Virtual Server)與真實伺服器(Real Server)，虛擬伺服器就是我們之前提過的負載平衡主機，又可以分成主要虛擬伺服器與備份虛擬伺服器，備份虛擬伺服器是主要虛擬伺服器的備份，當主要虛擬伺服器無法運作時，會取代主要虛擬伺服器執行它的功能，而真實伺服器就是伺服器叢集中的伺服器。以下是 Piranha 組成元件的說明：

- **pulse：類似之前介紹過的 heartbeat 軟體，用來監控主要伺服器是否正常。啓動 lvs。**
- **fos：由 pulse 啓動，負責 IP Takeover 功能，讓備份虛擬伺服器取代主要虛擬伺服器。**
- **send_arp：由 pulse 啓動，送出 ARP 廣播讓網路上主機更新虛擬主機的 MAC 位址。**
- **lvs.cf：lvs 的組態設定檔，定義了主要虛擬伺服器、備份虛擬伺服器、真實伺服器、負載平衡方式和排程演算法等所有伺服器叢集的相關資訊。可以由 Piranha 產生或手動建立。**
- **Piranha：用來建立 lvs.cf 的 Web 界面圖型化設定工具。**
- **lvs：由 pulse 啓動，讀取 lvs.cf 組態設定檔以設定伺服器叢集系統。包括呼叫 ipvsadm 建**

立虛擬繞送表(Virtual Routing Table)及呼叫 nanny 監控真實伺服器。

- **ipvsadm：由 lvs 或 nanny 呼叫，用來建立和修改虛擬繞送表。**
- **nanny：由 lvs 呼叫，用來監控真實伺服器，當真實伺服器無法運做時，會呼叫 ipvsadm 修改虛擬繞送表，把這個無法繼續提供服務的伺服器從伺服器叢集中移除。**



圖八、Piranha 的組成元件與彼此關係

接下來介紹 Piranha 的安裝流程：

1. **安裝相關軟體：**
   - kernel-2.2.16-3.i686.rpm：Piranha 0.4.16-3 需要 Linux kernel 2.2.16 以上版本
   - ipvsadm-1.11-4.i386.rpm：Linux Virtual Server(LVS)主程式 ipvsadm。
   - piranha-0.4.16-3.i386.rpm：Piranha 主程式，包括 fos、send_arp、lvs.cf、lvs 和 nanny。
   - piranha-gui-0.4.16-3.i386.rpm：Piranha Web 界面圖型化設定套件，用來設定 lvs.cf。
   - piranha-doc-0.4.16-3.i386.rpm：Piranha 的說明文件。
2. **設定伺服器叢集的伺服器的網路界面：**可以使用 ifconfig 或 linuxconf 等指令設定。
3. **啓動負載平衡：**有三種負載平衡方式可以選擇。

啓動 NAT：(在負載平衡主機設定)

a. 在 /etc/sysctl.conf 中加入 net.ipv4.ip_forward = 1 或執行 echo 1 > /proc/sys/net/ipv4/ip_forward。

b. 在 /etc/sysctl.conf 中加入 net.ipv4.ip_always_defrag = 1 或執行 echo 1 > /proc/sys/net/ipv4/ip_always_defrag。

c. 執行 ipchains -A forward -j MASQ -s n.n.n.n/type -d 0.0.0.0/0(n.n.n.n 是你的虛擬伺服器 IP 位址)。

啓動 IP Tunneling：(在每一台真實伺服器設定)

a. 執行 ifconfig tunl0 n.n.n.n up(n.n.n.n 是你的虛擬伺服器 IP 位址)。

b. 執行 echo 1 > /proc/sys/net/ipv4/conf/all/hidden。

c. 執行 echo 1 > /proc/sys/net/ipv4/conf/tunl0/hidden。

啓動 Direct Routing：(在每一台真實伺服器設定)

a. 執行 ifconfig eth0:0 n.n.n.n up(n.n.n.n 是你的虛擬伺服器 IP 位址)。

b. 執行 echo 1 > /proc/sys/net/ipv4/conf/all/hidden。

c. 執行 echo 1 > /proc/sys/net/ipv4/conf/eth0/hidden。

4. **設定 lcs.cf**：可以使用 Piranha Web 界面圖型化工具設定或手動建立。

5. **啓動 lvs**：執行/etc/rc.d/ini.d/pulse start 即可啓動 Piranha 伺服器叢集系統，它會呼叫 lvs。

6. 結論

伺服器叢集技術將會越來越重要，隨著上網人數的增加和網路技術的發展，當頻寬不足已不再是唯一可能的問題後，一個網站的效能和可靠度將成爲決勝的關鍵。另外，由於開放程式碼的影響，建立一個伺服器叢集有相當多的選擇，可以依造自己的需求和成本，選擇商業軟體或開放程式碼軟體，也可以選擇從各種單一套件手工打造或是使用整套叢集伺服器軟體。

參考資料

[1] Red Hat High Availability Server Project, "High Availability Server Project",

http://people.redhat.com/kbarrett/HA/

[2] David B. Ingham, Santosh K.Shrivastava, Fabio Panzieri, "Constructing Dependable Web Services", IEEE Internet Computing, Jan 2000

[3] Eric Dean Katz, Michelle Butler, Robert McGrath, "A Scalable HTTP Server The NCSA Prototype", http://www.ncsa.uiuc.edu/InformationServers/Conferences/CERNwww94/www94.ncsa.html

[4] Linux Virtual Server Project, "Linux Virtual Server Project", http://www.linuxvirtualserver.org/

[5] Cisco, "How to Cost-Effectively Scale Web Servers", http://www.cisco.com/warp/public/784/5.html

[6] The Eddie Mission, "The Eddie Mission", http://www.eddieware.org/

[7] The Backhand Project, "The Backhand Project", http://www.backhand.org/

[8] Linux-HA Project Web Site, "High-Availability Linux Project", http://linux-ha.org/

[9] Ultra Monkey, "Ultra Monkey", http://ultramonkey.sourceforge.net/