PC-Based 伺服器 Fail-Over 系統之建置
Realization of PC-Based Fail-Over System

林鳳銘　　吳守豪　　李蔡彥
國立政治大學電算中心
{l96in,swu,li}@nccu.edu.tw

摘要

網路上有越來越多的伺服器不僅需要提供每日 24 小時全年無休的服務，同時也不容許有當機或者暫時停機檢測的時候。例如我們定期都會收到不同系統管理者因執行例行性系統維護而需停止服務的 E-mail，這些中斷的服務通常需要數分鐘甚至到數天不等的時間才能恢復。這對亟需使用網路服務的使用者來說，是件十分不方便的事。因此，對這類的伺服器來說，如何建立一套 Fail-Over 機制便是一件刻不容緩的事。這裡的 Fail-Over 機制指的是當主伺服器出現問題時，另一台含有相同資料內容的 Fail-Over 伺服器能即時提供相同服務的方式。為了解決在伺服器無法正常運作時服務中斷的問題，我們利用 PC 及自由軟體元件建置了一套簡易而廉價的伺服器 Fail-Over 機制，可以在伺服器發生問題或者停機檢測時自動取代原主要伺服器的角色，繼續提供正常服務。對於發生故障的機器，也能在第一時間以簡訊等通訊方式通知系統管理者進行處理。我們以 LDAP 服務為例，說明此 Fail-Over 系統的可行性。

關鍵詞：網路服務、容錯機制、Fail-Over。

Abstract

There has been an increasing demand for providing 24-hour non-interrupted network services to users, even during the outage of regular system maintenance. It is common for us to receive this kind of notifications for maintenance that requires minutes or even days to resume the services. More and more users with urgent needs have experienced extreme inconvenience for this kind of service outage. Therefore, how to build a fail-over mechanism become a critical issue for system administrators. The fail-over mechanism in this paper means that a system with identical contents can take over the task of the primary server when it goes down for any reasons. We have designed a PC-based fail-over mechanism that can continue the network service when the primary server becomes unavailable. The system can inform the system administrator via short messages promptly when a system failure is detected. In addition, we will use the LDAP service as an example to illustrate the feasibility of such a fault-tolerant system.

Keywords: Network Service, Fault-Tolerant Mechanism, Fail-Over.

1. 前言

隨著網路的發達，使用者對網路服務（如電子郵件、LDAP 身份認證、DNS 主機名稱服務等）的要求，已如同水電等日常必需品一般，達到 24 小時全年無修的地步。這些服務如欲做到不因例行維修或不預期故障而間斷的目標，多需要透過容錯電腦系統的建置才能達成。目前有許多的廠商設計了一些不同的設備來執行 Fail-Over 的動作，然而這些設備的價格昂貴，動輒數十萬元，而且不同廠商的設備其設定方式也不盡相同，管理者必須學習不同的指令來管理這些不同廠商的設定。即使設定完成，實際上線的效能有些也未能達到廣告所宣稱的效果。同時該設備對管理者來說像是一個黑盒子(black-box)一般，當發生問題時，只能求助於生產該設備廠商的工程人員，並不能立刻解決伺服器無法上線的問題。

在以自由軟體自行建構系統方面，有專文探討 ADSL 線路的負載平衡。例如，[1]中主要是針對兩條不同的 ADSL 線路來執行負載平衡，可以達到兩條線路都有資訊流通，不會造成流量都往其中一條線路擠的情形。不過[1]並未對其中一路發生問題時，會不會造成有些封包會丟給發生障礙的那一路線路的狀況提出解決方式。[2]提出了針對特定服務 web proxy 所連接的 ADSL 線路來執行負載平衡。然而，這個方式僅適用於特定服務所連結的 ADSL 線路之負載平衡。[2]中也並未對當其中的 某些 ADSL 線路出現障礙時，系統會不會發生將流量導入發生異常的 ADSL 線路，導致使用者覺得網路斷線的情形提出解決方案。

容錯機制(Fault-Tolerance)是為了提升系統可使用度(Availability)而經常與負載平衡被相提並論一種技術。相關文獻及產品中，提供容錯機制的方式，根據所使用的軟硬體技術及容錯要求的不同而分成許多種。傳統的容錯系統設計多使用在擔任重要任務(mission critical)的大型主機上。但是隨著軟硬體技術的發展，PC 等級的方案也逐漸成為開發的焦點[3]。

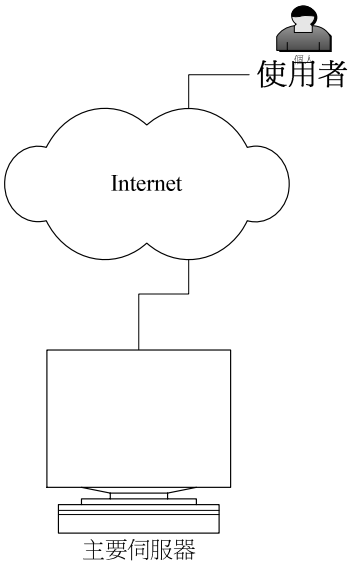

圖 1. 正常的伺服器連結狀態

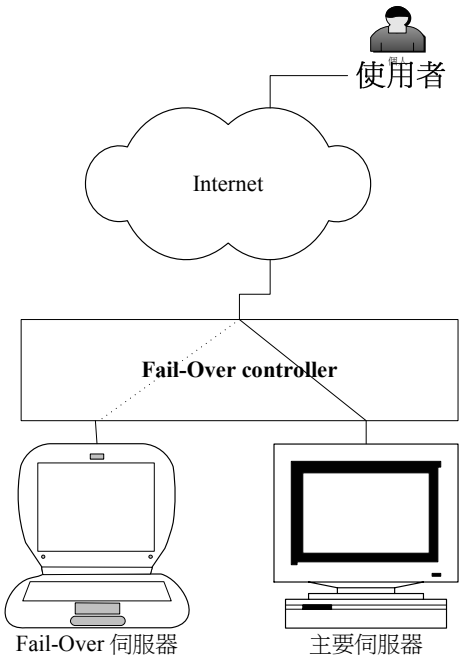

圖 2. Fail-Over 架構

最常見的容錯系統是以多重硬體(Hardware Redundant)的方式，配置一個以上的電腦元件，以提供即時的線上備援。其中，又以提供即時的多重備援儲存裝置所需的技術最複雜。這些儲存裝置除了需要有共享的資料匯流排外，還需作業系統的配合(如 MS Windows Clustering[9])，才能達到即時線上備援的目標。這類方式的優點是能提供最細緻的容錯及完整的線上備援，而缺點則是價格過高。另一類常用的方式是以較高階的網路設備（Layer4 交換器或防火牆）或提供容錯作業系統的主機，置於網路服務的前端做為 Proxy，根據伺服器的狀態前轉使用者的網路封包到最適當的主機。另外一類則是在作業系統之上，純粹以軟體技術做到 Fail-Over 的機制，如[7]所發展的軟體。這類機制的缺點是 Fail-Over 的發生的時效較差（通常需數秒鐘），因此在發生問題當時所提出的網路服務要求將會失敗，而有賴使用者再提出一次新的要求。如果所提供的服務包含動態的資料查詢，則此 Fail-Over 的機制通常還需要有在機器間複製資料(Data Replication)的功能。最後，對一個分散式系統而言，也可以在單一應用程式的範圍內設計具有容錯功能的軟體，然而此類系統不在本論文所探討的範疇內。

容錯機制的選擇或開發，除了受經費預算的影響外，還需依照網路服務的特性及使用者的需求度來設計。目前各單位普遍的現象都是伺服器越來越多，當需要某些特定的服務時就建立一部新的伺服器來提供該服務；而有些服務的提供，必須仰賴使用者登錄方能使用特定的資源。因此，伺服器的增加代表了使用者必須記住的密碼也隨之增加。為了解決記憶多組不同密碼的不便，我們導入了 LDAP 伺服器來做為密碼統一驗證的伺服器。由於該伺服器管理了大部分主機的使用者名稱與密碼，相形之下該伺服器的重要性也與日遽增，同時由於多個系統的使用者名稱與密碼驗證都透過該伺服器來認證，為了讓使用者能夠在該機器進行例行性系統維護時仍能繼續使用網路資源，我們建立了一套伺服器 Fail-Over 系統，來讓 Fail-Over 伺服器在主 LDAP 伺服器短暫停機維護時，取代主 LDAP 伺服器，繼續提供密碼驗證的服務。

我們根據網路應用的特性，設計了一個建構在自由軟體及一般 PC 之上，純粹以軟體技術達到的廉價 Fail-Over 系統。此 Fail-Over 系統，不僅適用於我們目前所使用的 LDAP 伺服器，對於使用者資訊不是即時變動的伺服器，如 DNS、WWW 等而言也都可以適用。

2. 系統架構

圖 1 所示是正常的網路連結方式，當使用者想要擷取伺服器提供的服務時，使用者透過網際網路直接連接想要連結的伺服器，以進一步取得想要的資訊。圖 2 則是我們提出的架構。根據圖 2，我們改變了原來的架構，讓使用者並非與提供服務的伺服器直接連結，而必須透過我們所提出的 Fail-Over controller 來執行封包轉送的任務。對使用者而言，將不會發現到必須事先連接到 Fail-Over controller，然後再由 Fail-Over controller 根據伺服器的狀態決定該與哪一部伺服器連結。

2.1 Controller 系統架構

如圖 3. 對 Fail-Over 系統架構而言，主要包含 Fail-Over controller 與一部 Fail-Over 伺服器。Fail-Over controller 的主要工作是透過 watch dae-

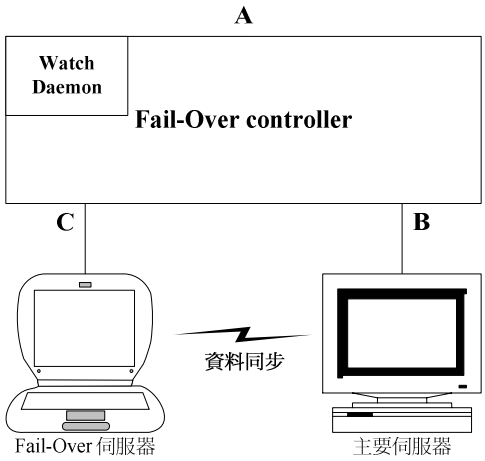

圖 3. Fail-Over controller 架構

mon 來隨時監測 A 點的網路流量，同時根據網路流量資訊來確認主要伺服器的運作情形。在正常的情況下，A 點的網路流量都會流到 B 點，然後送給主要伺服器。當發生異常情況如主要伺服器進行系統維護或者主要伺服器發生異常無法繼續提供服務時，watch daemon 將會及時發現此一現象，同時 watch daemon 會執行重新建構 Fail-Over controller 的動作，重新建構過的 Fail-Over controller 就會將 A 點的網路流量導到 C 點給 Fail-Over 伺服器。對使用者而言這將只是短暫的斷線，同時使用者也不容易感覺到他已經到連結 Fail-Over 伺服器。這個短暫的時間就是 watch daemon 偵測主要伺服器無法提供服務，然後重新將網路點 A 的流量導到網路點 C 的時間。

一個 Fail-Over controller 可以控制一組以上的網路服務，並對不同的服務提供不同的健康檢測功能及封包轉向服務。但為避免此 controller 成為另一個可能的故障點或流量瓶頸，因此在實際運作上，對所提供的 Fail-Over 服務數量仍須有所控制。

2.2 流量監測

對一般的具 Fail-Over 功能的容錯系統而言，最常見的運作模式是以 Heart-beating[8]的方式檢視主要伺服器的健康狀態，藉以決定是否啟動次要伺服器的服務。然而，以大部分 Heartbeating 的偵測模式為例，大多只能知道通往該伺服器的網路是否依然暢通；對於網路正常但特定的網路服務程式已經故障的情況而言，傳統 Heartbeating 的檢測是無濟於事的。因此，我們在此論文所實做的系統中，以兩階段的模式進行網路服務的健康檢測。第一階段是借用作業系統中網路轉址的工具，瞭解主要伺服器的活動狀態。當此伺服器的活動停止時，才啟動第二階段依應用程式客製化的 Heart-beating 程式，做進一步的檢測工作。

就圖 4 而言，在 Fail-Over controller 中最重要的工作是 watch daemon 隨時監測 A 點的網路流

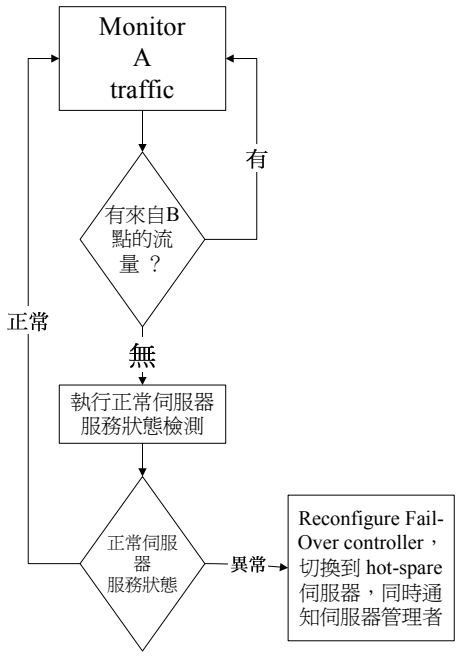

圖 4. watch daemon 架構

量，然後根據所收集的資料，決定封包的走向。這裡我們將對 watch daemon 的運作方式作一說明。在圖 3 中 watch daemon 不斷地接收來自 A 點的網路流量，然後他會檢測收到封包的來源端位址是否有來自 B 點的網路流量。如果收到封包的來源位址有來自 B 點所送出的封包，表示主要伺服器的運作沒有問題，他就會繼續執行收集來自 A 點網路流量的工作。當他收集到的封包並沒有來自 B 點的網路流量時，他就會執行額外的主要伺服器服務狀態檢測程序，然後根據伺服器服務狀態的檢測結果來決定是否要改變封包的走向。當伺服器服務狀態的檢測結果為正常時，他將會返回，同時回到執行收集來自 A 點網路流量的工作的狀態，當伺服器服務狀態的檢測結果為不正常時，他會重新建構 Fail-Over controller ，把來自 A 點的封包轉送給 Fail-Over 伺服器，並且將這個異常的結果透過簡訊傳送給伺服器的管理者。

2.3 資料同步化

Fail-Over 伺服器是一部完整的伺服器，但它不一定要與主要伺服器的等級相同或者完全一樣的伺服器。只要能夠在主要伺服器發生異常時，暫時提供服務即可。然而此伺服器也不能使用等級差太多的機器，以免在主要伺服器發生狀況，Fail-Over 伺服器接手提供服務後不久就因承受不了網路負載而發生問題。也就是說 Fail-Over 伺服器的等級與主要伺服器的負載成正比。由於 Fail-Over 伺服器的角色就是一部與主要伺服器相同資料的伺服器，所以 Fail-Over 伺服器必需每隔一段時間就與主要伺服器做資料同步的工作。在本論文所實做的

系統中，我們使用 rsync[3] 這支程式來做 Fail-Over 伺服器與主要伺服器間的資料同步工作，由於 rsync 可以針對有異動的檔案才執行資料同步的動作，因此以資料異動性不大的 LDAP 伺服器為例，資料同步的週期毋需太短；目前我們將資料同步的時間區間訂為為十分鐘。

2.4 伺服器異常訊息之通知

一般而言，對於網路或是伺服器的異常狀況通常透過 E-mail 或是網頁的方式來呈現，網路管理者或伺服器管理者需要上網接收 E-mail 或者查詢網頁資料來得知網路或伺服器的狀況。在網路暢通或者管理者可以接觸到電腦的情況下，可以方便的知道並處理網路或伺服器的異常狀況，不過當管理者所處的環境沒有網路或電腦時，就不容易及時獲知網路或伺服器的異常狀況。因此，在本論文所實做的系統中，我們為 Fail-Over 架構建立了簡訊發送機制，透過這個方式，網路或伺服器的管理者就可以透過另一類網路的服務，隨時掌握網路或伺服器的狀況。

以本論文所實做的系統為例，我們安裝一個 short message service gateway[5]，然後透過這個 gateway 將我們想傳送的訊息傳送出去。

3. 系統實作

在上一節中，我們提出了一個適用於自由軟體等不同 UNIX 系統的 Fail-Over 架構。在這一節中我們將對上一節所提出的架構，說明他們的實作方式。

對於國立政治大學而言，我們有許多的驗證帳號的方式都是透過 LDAP 伺服器來完成。但由於 LDAP 服務的重試機制(retry)是掌握在 Client 端的應用程式，因此即使我們建置了兩台內容相同的伺服器，也無法保證所有的 Clients 在主要伺服器故障時，能使用次要伺服器的備援服務。所以我們需要對 LDAP 伺服器建立一個 Fail-Over 機制。我們使用了一部 XEON PC 伺服器來當作 Fail-Over controller，對於作業系統我們則是使用了 Freebsd 5.X [7]。其運作方式如下：

1. 在 作業系統 方面我們重新 compile kernel，同時加入了 ipfilter 的功能，使得我們可以在 kernel level 做封包的轉送。
2. 我們以 perl 來實作 watch daemon，這個程式會不斷的觀測封包的進出情況，同時根據封包的進出情形來決定封包需要轉送給主要伺服器或是轉送給 Fail-Over 伺服器。
3. 對於資料同步方面，我們使用 rsync 這支程式來達到動態資料同步的工作。
4. 當異常狀況出現時，我們建立的簡訊發送機制，會將異常的狀況傳送給管理者。

目前這個系統為正在上線提供正常服務的系統。以 2004/08/06 為例，此 LDAP 系統的負載為 4.46 connections/sec。而本論文所提議的 Fail-Over 機制，在主要伺服器發生故障時，已能將服務導向 Fail-Over 伺服器，繼續提供正常的服務。

4. 結論

我們所建立的伺服器 Fail-Over 系統優點在於可以以簡易而廉價的系統建立無中斷的伺服器服務。對使用者而言，將不容易感受到系統切換的短暫斷線狀態。這個系統已可以達到全年無休伺服器的概念。如前所述，對於不是以硬體實做的容錯伺服器而言，使用者一段週期時間所面對的暫時性斷線將是不可避免的情況。

對管理者而言，必須時時確保伺服器的健康狀態，以防止一旦出現伺服器異常的狀況時，對使用者所造成的影響。當出現問題時，管理者必須盡快修復，然而在問題發生的當下，管理者的所承受的壓力是非常大的。尤其是一些比較關鍵的機器出問題時，管理者必須盡最大的努力讓該伺服器盡快恢復服務。有了這個 Fail-Over 機制，管理者可以在 Fail-Over 伺服器取代原有主要伺服器提供服務時，從容的檢修需要維護的機器。

本文中提出的伺服器 Fail-Over 系統中資料的同步問題還有待進一步的改進，就目前來說是以十分鐘為單位來執行同步的工作，也就是說當主要伺服器發生異常的狀況，同時透過 watch daemon 將資料轉給 Fail-Over 機器時，會有一小段資料不同步的情況，這點還有進一步的改良空間。另外，目前本系統對服務中斷後的處理，尚未提供自動復原的機制；其主要原因在於故障情況是否已排除仍難自動偵測，自動復原機制可能造成 Fail-Over 的情況重複一直發生。因此，人為介入確認服務的復原仍是較為妥善的方式。

參考文獻

[1] 金志誠，"利用無硬碟 PC 建置的負載平衡系統"，TANET 2003 研討會論文集，pp. 1-4。

[2] 范修維、廖鴻圖、張維世，"Web proxy 負載平衡器建置策略"， ，TANET 2002 研討會論文集，pp. 1017-1022。

[3] P. Lewis, "Strictly On-Line: A High-Availability Cluster for Linux," *Linux Journal*, 1999. Available: http://www.linuxjournal.com/article.php?sid=3247

[4] http://samba.anu.edu.au/rsync/.

[5] http://www.freebsd.org/

[6] http://www.kannel.org/.

[7] http://www.legato.com/

[8] http://www.linux-ha.org/heartbeat/

[9] http://www.microsoft.com/windows2000/technologies/clustering/