

# 採用 RR-DNS 機制的網路服務可用性之研究

## On the Availability of Network Services Based on RR-DNS

蕭淵隆\* 陳彥鋒\*\* 黃景煌\*

\*國立暨南國際大學 資訊工程學系

\*\*國立暨南國際大學 資訊管理學系

{s9321020, ycchen, hching}@ncnu.edu.tw

### 摘要

循環式領域名稱服務(Round-Robin DNS, RR-DNS) 是目前網際網路上常用的服務負載平衡機制, 服務提供者可利用多台伺服器, 藉由共用同一個主機名稱來達到平均分擔負載的目的, 然而任何一台伺服器故障均會導致部份使用者無法順利存取服務。使用 RR-DNS, 網路服務的可用性並沒有因為伺服器的增加而提高。由於各式的網路應用越來越重要, 高可用性網路服務需求日益殷切, 本篇論文提出一個利用現有 RR-DNS 提高網路服務可用性的機制, 此機制定期監測各伺服器, 判斷伺服器是否能正常提供服務, 並針對有問題的伺服器, 立即更改 DNS 伺服器的設定, 以避免使用者存取有問題之伺服器。藉由維護 DNS 的主機列表, 我們可以使用多台伺服器, 不僅提供負載平衡, 也可提高服務的可用性。

**關鍵詞:** Round-Robin DNS, 可用性, 負載平衡, 網路管理

### Abstract

Round-Robin DNS (RR-DNS) is an easy and popular way to provide load balance for Internet services. By having multiple servers with the same host name, the service load can be equivalently shared by the servers. However, the breakdown of any server in the server group may lead to access failures for a portion of users. Indeed, the use of RR-DNS cannot increase the overall availability of the service even more servers are deployed. Due to the importance of network applications, there is a strong demand for high availability of network services. In this paper, we propose an effective approach to provide a high available network service based on RR-DNS. All the servers are monitored periodically. When a server fails, the IP address of the server is temporarily removed from the round robin address list for the corresponding host name. Therefore, the access of the failed server will be prevented. As a result, our approach can increase the availability of a network service by increasing the number of servers.

**Keywords:** Round-Robin DNS, Availability, Load balancing, Network management

### 1. 前言

領域名稱服務(DNS)[5]幾乎是所有網路服務上必備的基礎服務之一, 所有只要需要轉換網域名稱到 IP 位址上的服務都是要經由 DNS 來處理。近年來網際網路發展迅速, 各式網路服務的需求增加, DNS 服務隨著網路成長, 產生許多的變形; 例如可以處理動態 IP 位址的動態網域名稱 DNS, 以及應用於一個網域名稱對應於多個 IP 位址的循環式(Round-Robin) DNS (簡稱 RR-DNS)[3], 以及其他功能不盡相同的 DNS 服務。本篇論文想探討目前採用 RR-DNS 機制的網路服務所面臨的問題, 以期提出改進的方法。

由於現今的網路需求越來越高, 各式各樣的服務需求量也跟著提高, 以往使用單一伺服器便可處理所有使用者需求的運作方式已漸不可行。為有效減輕伺服器的負載, 同時使用多台伺服器提供一項服務已成為常見的網路服務架構, 使用具負載平衡功能的路由器、電腦叢集(Cluster)、多處理器系統、以及 RR-DNS 都是可以用來改善系統負載的方式。這些方法可以獨立使用或同時並行、或專為 WWW 服務設計負載平衡機制[4]。在眾多的平衡機制中, RR-DNS 是最簡單且經濟的網路服務負載平衡機制。所謂 RR-DNS 是將多台伺服器的 IP 位址都對應到同一個網域名稱, 當用戶端至 DNS 伺服器詢問該網域名稱時, DNS 伺服器會以循環的方式提供不同優先順序的 IP 位址資料, 使得來自用戶端的服務要求平均地分配至每一台伺服器上, 達到負載平衡的目的。這種利用 RR-DNS 的負載平衡機制只把服務要求平均地分配至每一台, 無法保證真正的負載平衡, 不過採用 RR-DNS 只需對 DNS 做簡單的設定, 不需要任何額外特別的硬體, 因此是目前網路上很常見的負載平衡方式。

對於網管人員而言, 採用 RR-DNS 時所要面對的挑戰是當伺服器數量增加, 如何確保每一台機器隨時都在正常運作。在實務上, 伺服器從故障, 經發現至修復, 通常會延續一段時間, 因此任何一台伺服器故障均會導致網路上部份使用者無法順利存取服務。使用更多的伺服器固然可以降低每台伺服器的負擔, 網路服務的可用性雖有可能因伺服器的負擔降低而減少故障的機率而有所改進, 但是伺服器數量增加不易於管理, 反而有可能不利於網路服務的整體可用性。此外, 近年來由於網際網路蓬

勃發展，人們對於網路服務的品質要求越來越高，對於網路服務的高可用性(High Availability, HA)需求也日趨增多，因此如何使用簡單的 RR-DNS 機制，藉由增加伺服器數目提高網路服務的可用性是本論文的主要研究課題。所謂可用性，對於主從式的網路服務而言，我們可以使用在一段長的時間內，成功存取服務的次數佔所有服務要求次數的百分比來衡量，或是以可提供服務的時間佔所有時間的百分比[6]。本論文提出一個動態調整 RR-DNS 來提高網路服務可用性的調適式 RR-DNS 機制，主要的方式是利用一個伺服器狀態監測程式，定期地對伺服器群中的每一伺服器進行狀態監測工作，當發現有任何無法正常提供服務的伺服器，該監測程式會自動至 DNS 伺服器修改 DNS 設定，暫時將對應的 IP 位址移除，使得故障的機器得以隔離，直到恢復正常後，再重新將其 IP 位址加入 DNS 設定。此機制對企業內部網路服務，可有效提升服務的可靠度，對於網際網路服務，其所能提升的可用性會因遠端 DNS 伺服器的快取(Cache)因素而有所影響，可以利用降低 DNS 的 TTL(Time To Live)值來改進。此外，伺服器監測程式的監測週期，也影響 DNS 動態調整的及時性。我們將於論文中探討分析這些因素，並以模擬實驗驗證我們的想法。

本篇論文其他章節分述如下，第 2 節介紹 RR-DNS 原理，第 3 節描述調適式 RR-DNS 機制及其伺服器狀態監測程式，第 4 節為可用性分析，並探討影響可用性的因素，第 5 節為結論與未來進一步研究的方向。

## 2. RR-DNS 原理

在有些網路服務應用上，我們可能會隱藏真正提供同一服務的多台主機，而以一個單一的領域名稱代表，以方便使用者記憶並可實現簡單的負載平衡，要達成這個目的最簡單的方法即是利用循環式的 DNS 技術，即 RR-DNS。一般的 DNS 服務是單一的領域名稱與 IP 位址對應，使用 RR-DNS，單一的領域名稱對應至一組 IP 位址。例如，假設我們有 3 台主機，IP 位址分別是 10.0.0.1，10.0.0.2 及 10.0.0.3，我們選擇 proxy.mysite.com 做為此 3 台主機共同的領域名稱，則在 DNS 伺服器的部分區域設定檔案(zone file)[2]內容如下：

```
@ IN SOA ns.mysite.com. nsmaster.mysite.com. (
    2003081403 ; Serial
    28800 ; Refresh
    14400 ; Retry
    3600000 ; Expire
    86400 ) ; Minimum

IN      NS      ns.mysite.com.
proxy  IN      A      10.0.0.1
        IN      A      10.0.0.2
        IN      A      10.0.0.3
```

當有三個用戶端 client1，client2，及 client3 依序查

詢 proxy.mysite.com 的 IP 位址時，DNS 伺服器會以循環(即 round-robin)的方式回應相對應的 IP 位址串列(list)。以下以 nslookup 指令為例，查詢及回應內容如下：

```
client1> nslookup proxy.mysite.com
Server: ns.mysite.com
Address: 192.168.0.1
Name: proxy.mysite.com
Addresses: 10.0.0.1 10.0.0.2 10.0.0.3
```

```
client2> nslookup proxy.mysite.com
Server: ns.mysite.com
Address: 192.168.0.1
Name: proxy.mysite.com
Addresses: 10.0.0.2 10.0.0.3 10.0.0.1
```

```
client3> nslookup proxy.mysite.com
Server: ns.mysite.com
Address: 192.168.0.1
Name: proxy.mysite.com
Addresses: 10.0.0.3 10.0.0.1 10.0.0.2
```

我們可以看到每個用戶端所得到的位址串列都包含三個 IP 位址，然而位址出現順序不同，是以循環的方式依序出現。目前大部的用戶端應用程式多只使用 DNS 伺服器回傳的 IP 位址串列之第一個出現的 IP 位址。因此，三個用戶端的服務請求便被平均地分散至三台主機了。使用 RR-DNS 的具有以下優點：

- (1) 無需增加成本：大部 DNS 系統均有支援 RR-DNS 功能。
- (2) 容易設定：僅需更改少部分的 DNS 記錄資料。
- (3) 簡易的負載平衡：藉由循環排列出現的 IP 位址達到依服務請求順序平均分配伺服器達到簡易的負載平衡功能。

RR-DNS 的具有以下缺點：

- (1) RR-DNS 的簡易負載平衡其實僅能做到負載分享(load sharing)無法做到真正的負載平衡。
- (2) DNS 伺服器無法自動偵測後端各服務主機的狀態，如果某台主機故障，除非手動地將伺服器從 DNS 表格中移除，否則用戶端仍有機會存取到這台故障的主機。
- (3) 許多 DNS 伺服器會暫存先前的查詢結果(即 Cache 機制)，直到 TTL 時間逾期，因此 zone file 的變更需要一段時間才能反應到整個網際網路。

## 3. 調適式 RR-DNS 機制

為了有效提升網路服務的可用性，我們必須儘可能讓用戶端避免存取已經發生障礙的伺服器，由於用戶端在存取網路服務前，會至 DNS 伺服器詢問該網路服務領域名稱之 IP 位址，只要我們能在發現伺服器障礙時，能夠及時變更 DNS 的相關設

定，讓用戶端在詢問 DNS 時所得的資料只包括目前正常運作的伺服器之 IP 位址，那麼就有可能降低用戶端連結到故障伺服器的機率，進而提高網路服務的整體可用性。為提供以上的動態調整 RR-DNS 的功能，首先我們需要一個伺服器狀態監測程式，定時地至各伺服器監測，了解各伺服器是否正常運作。我們可以使用伺服器本身所提供服務來監測，例如伺服器主要提供 WWW 服務，我們可以使用 HTTP 協定嘗試與之連結，甚至可讀取某一測試用網頁來確定網站是否正常運作，若為一般用途伺服器我們可以使用 ping 程式來判斷伺服器是否仍活著。

圖 1 顯示此監測機制之系統架構，以圖 1 為例，狀態監測程式(Status Monitor)是以週期性輪詢(Polling)方式至各伺服器監測其狀態。狀態監測程式可以負責多個伺服器群的狀態監測工作，每個伺服器群的監測週期可依可用性需求高低分別調整，有關監測週期與可用性的關係，可參考第四節的分析。在一個輪詢週期內，當狀態監測程式發現某一伺服器無法正常提供服務時，例如領域名稱為 svr.ncnu.edu.tw 的伺服器#2 發生障礙，狀態監測程式便會立即至 DNS 伺服器將該伺服器(即伺服器#2)的 IP 位址從所屬領域名稱(svr.ncnu.edu.tw)的 RR 位址串列中暫時移除。類似的情形，當狀態監測程式發現原先故障的伺服器恢復正常時，便會再至 DNS 伺服器將恢復正常的伺服器之 IP 位址重新加入原 RR 位址串列中。這種動態調整 RR-DNS 的機制，對於網路服務可用性的提升，依用戶端所在網路領域而有所不同。

(1). 用戶端與網路服務位於同一網路領域：

用戶端與網路服務位於同一網路領域的情形，如企業內部網路服務，或供校內師生使用的校園網路服務。這些位於近端的用戶端，每次存取網路服務前，均會至直接管轄該網路服務的 DNS 伺服器詢問 IP 位址。基本上 DNS 伺服器的 RR 位址串列資料的即時性等於狀態監測程式所採用的輪詢週期，亦即，如果狀態監測程式每一分鐘輪詢伺服器群一次，則 DNS 伺服器的 RR 位址串列資料的正確性會有一分鐘的誤差。因此，有可能某一伺服器在發生故障一分鐘後，其 IP 位址才從 DNS 伺服器的 RR 位址串列中移除，也因此在這一分鐘內，便有可能有用戶端因企圖連結至該伺服器而導致存取失敗的情況發生。顯然，狀態監測程式的輪詢週期越短，存取失敗的情況便會越少發生，網路服務的可用性便會越高。因為發生障礙的 IP 位址愈早被排除，其後的服務要求便會自動分派至正常運作的伺服器，整體的可用性因成功的服務存取次數增加而提高。

(2). 用戶端與網路服務位於不同的網路領域：

當用戶端與網路服務位於不同網路領域時，例如存取 Internet 上的服務時，用戶端會先至自己所屬領域的 DNS 伺服器詢問，如果此 DNS 伺服器沒

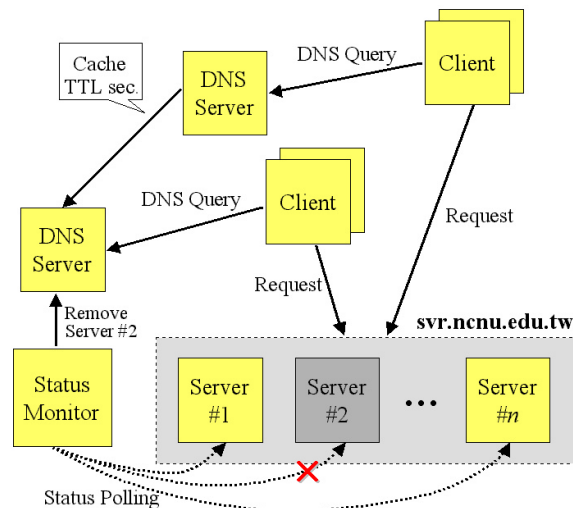


圖 1 調適式 RR-DNS 機制系統架構

有所詢問的資料，它會沿著領域名稱階層找到提供 IP 位址對應資料的 DNS 伺服器，以獲得目前對應某一領域名稱的 RR 位址串列資料，此位址資訊除了提供給用戶端外，並以快取方式存於 DNS 伺服器，以便快速提供查詢資料予其他用戶端。DNS 伺服器存放 IP 位址對應資料的快取存活時間可由 DNS 的 TTL 參數決定。類似前述，如果 TTL 值為 30 分鐘，有可能某一伺服器在發生故障 30 分鐘後，其 IP 位址才從遠端的 DNS 伺服器的 RR 位址串列中移除，同樣地，在這 30 分鐘內，會有許多用戶端因 DNS 伺服器的 TTL 時間尚未到達，而拿到舊的 RR 位址快取資料，因而企圖連結至故障的伺服器，導致存取失敗。因此遠端 DNS 伺服器反應服務狀態的即時性會有 TTL 時間的延遲，此延遲影響服務的可用性。顯然，TTL 越小，存取失敗的情況相對減少，網路服務的可用性便會升高。因此，在要求提高服務的可用性前提下，我們可考慮把 TTL 值調小。不過，必須小心 TTL 值太低將使得詢問 DNS 伺服器的頻率過高，影響網路效能與 DNS 伺服器負擔。

#### 4. 可用性分析

假設伺服器的數目為  $n$ ，每台伺服器的可用性均為  $A$ ，則原始 RR-DNS 機制所提供網路服務之可用性：

$$A_{rr} = A \tag{1}$$

因為在原始 RR-DNS 機制下增加伺服器的數目只能降低伺服器的負載，並無法提高系統的可用性。使用調適式 RR-DNS 機制時，倘若每當伺服器故障時，DNS 便能自動調整，則只有當所有伺服器都故障時，才有可能導致使用者無法存取服務，因此調適式 RR-DNS 機制在理想情況下的可用性為：

$$A_a = 1 - (1 - A)^n \quad (2)$$

以下我們先針對在沒有 TTL 影響的近端網路服務存取進行可用性分析。假設  $T_p$  為狀態監測程式的輪詢週期，而  $T_r$  為伺服器故障的平均恢復時間， $T_p$  必須小於  $T_r$ ，狀態監測工作才有意義。每當有一伺服器發生故障時，一般而言，我們在其故障至恢復的  $T_r$  時間中的前  $T_p$  時間內，可能會因狀態監測程式尚未發現障礙，而無法適時調整 DNS 設定來提高服務的可用性，因此這段期間網路服務的行為模式與與原始 RR-DNS 機制並無二致，因此，相較於理想情況的調適式 RR-DNS 機制，實際的調適式 RR-DNS 機制，有  $T_p/T_d$  比例的時間內維持原始的 RR-DNS 機制，其他比例的時間方有調適式 RR-DNS 功能，因此，調適式 RR-DNS 機制實際的可用性為：

$$A_p = (T_p/T_d) \times A_{rr} + (1 - T_p/T_d) \times A_a \quad (3)$$

從以上公式可看出，如果  $T_p$  趨近於 0，即當一有伺服器故障隨即更新 DNS 設定，則  $A_p$  將趨近於理想情況的可用性  $A_a$ 。如果  $T_p$  大到等於或超過  $T_d$ ，即當有伺服器故障至其恢復時，狀態監測程式都還未發現，那麼 DNS 設定就幾乎不會改變，與原始 RR-DNS 機制是一樣的，因此  $A_p$  就會趨近於  $A_{rr}$ 。圖 2 顯示  $A_{rr}$ 、 $A_a$  與  $A_p$  之間的關係與以及各可用性值與伺服器數目  $n$  的關係，圖 2 之分析是假設每一台伺服器的可用性為 0.95，輪詢週期為一分鐘，而伺服器平均恢復時間為 10 分鐘，我們可以看出採用調適式 RR-DNS 對可用性有明顯的提升，此外，我們可以發現當伺服器數目增加至 4 台時，可用性便能達到 99.5%。

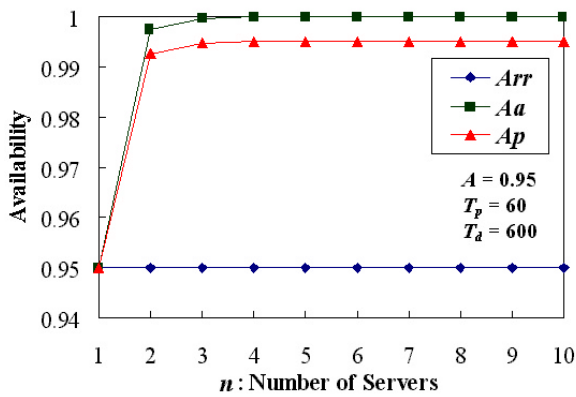


圖 2 可用性與伺服器數目之關係

另外，圖 3 顯示輪詢週期對可用性的影響，隨著  $T_p$  從 30 秒、1 分鐘、增加至 5 分鐘，我們可以看出網路服務的可用性越來越小，依據公式(3)，網路服務的可用性隨著輪詢週期的增加呈現線性遞減趨勢，因此，在監測程式與網路的負載之承受範圍內，我們可以儘可能地縮短輪詢週期，以提高服務的可用性。

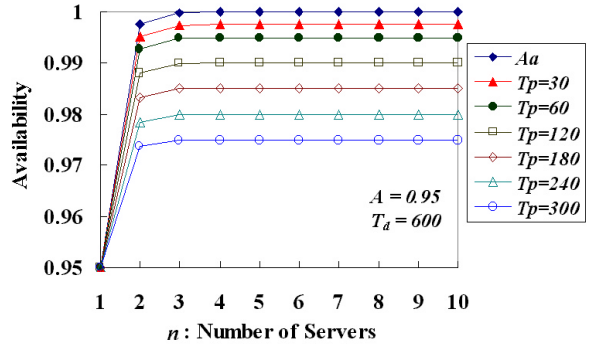


圖 3 輪詢週期與可用性之關係

以上的分析並沒有考慮 DNS 的 TTL 參數，因此屬於近端或企業內部網路的服務可用性分析。由於 TTL 值( $T_{tl}$ )通常大於  $T_p$ ，對遠端的用戶而言，近端的 RR-DNS 雖然在  $T_p$  時間動態調整，卻無法隨即將更新資訊傳達至遠端的 DNS 伺服器，而必須在 TTL 逾期時，由遠端 DNS 伺服器讀取最近的 IP 位址對應資料。因此每當有伺服器發生故障時，在其故障至恢復的  $T_d$  期間，發生故障  $T_p$  時間後，近端 DNS 伺服器設定將會更新，而遠端 DNS 伺服器會在  $T_{tl}$  時間後才會更新。如果  $T_{tl} > T_d$ ，則伺服器從故障至恢復，遠端 DNS 伺服器有可能完全沒有任何變動，因此，調適式的 RR-DNS 機制在當  $T_{tl} > T_d$  時並沒有任何作用，可用性與原始 RR-DNS 機制相同。如果  $T_{tl} \leq T_d$ ，調適式 RR-DNS 機制有  $T_{tl}/T_d$  比例的時間內維持原始的 RR-DNS 機制，其他比例的時間方有調適 RR-DNS 功能，因此，對於遠端用戶，調適式 RR-DNS 機制之可用性為：

$$A_t = \begin{cases} A_{rr} & , T_{tl} > T_d \\ (T_{tl}/T_d) \times A_{rr} + (1 - T_{tl}/T_d) \times A_a & , T_{tl} \leq T_d \end{cases} \quad (4)$$

圖 4 顯示 TTL 值與可用性的關係，TTL 值增加，可用性值會隨之變小，此現象類似於輪詢週期對可用性的影響，因此，在 DNS 伺服器與網路負載可承受範圍內，我們可以儘量使用較低的 TTL 值來增進遠端 DNS 伺服器資料的即時性，進而改善服務之可用性。

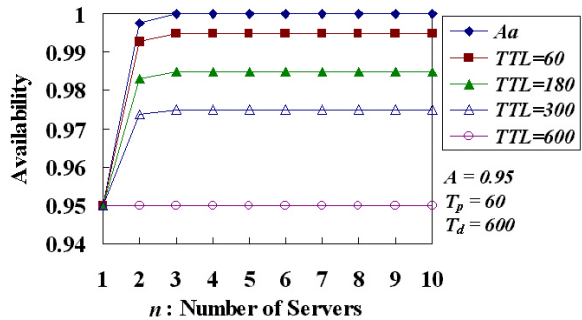


圖 4 TTL 值與可用性之關係

歸納以上的分析，我們發現降低輪詢週期及 TTL 值均可提高服務的可用性，然而在同時考量輪詢週期與 TTL 值對可用性之影響時，如果我們只考慮遠端用戶所獲得的可用性，在給定的 TTL 值下，低於 TTL 值的輪詢週期無助於可用性的提升，因為每一個輪詢週期更新一次的近端 DNS 設定必須等待 TTL 時間才能反應於遠端的 DNS 伺服器。此外，在給定的輪詢週期下，低於輪詢週期的 TTL 值也無助於可用性的提升，因為遠端 DNS 伺服器頻繁地至近端 DNS 伺服器所詢問的資料，最快也要一個輪詢週期的時間才更新一次。因此，在不考量其他的環境因素下，我們建議輪詢週期與 TTL 兩者的值應該相同，以避免任一方無謂的資源浪費。

## 5. 結論與未來方向

如何提高服務的可用性一直是個重要課題，採用專為提供高可用性服務所設計的軟硬體架構固然是有效的解決方案，然而所費不貲，本論文探討如何利用現有的 RR-DNS 機制來提升網路服務的可用性，我們所採用的調適式 RR-DNS 機制，無需任何特殊的軟硬體，只需使用一般的伺服器執行簡單的服務狀態監測程式即可。因此，網管人員可以很容易利用現成的設備建置具有較高可用性的網路服務。

我們針對調適式 RR-DNS 機制的可用性進行分析，發現增加伺服器數目的確可以提升服務的可用性，例如使用 4 台可用性為 95%、平均恢復時間為 10 分鐘的伺服器，便可將服務的可用性提升至 99.5% 以上。此外狀態監測程式的輪詢週期與 DNS 伺服器之 TTL 值也會影響服務的可用性，分析結果顯示兩者的值越小越好，且最好能夠相同，如此方能以最有效的方式提升服務可用性。

本論文所提調適式 RR-DNS 機制是依據伺服器是否正常來動態調整 DNS 伺服器設定，未來我們希望能夠進一步監測各伺服器中的 CPU 負載或可用資源，並依據監測結果動態調整 DNS 伺服器的 RR 位址串列之 IP 位址出現順序與機率，進而精確地依照伺服器負載來分配用戶端的需求，達到真正的負載平衡之目的。也可針對負載超過預設臨界值的伺服器，在還未危害服務可用性前，暫時從 RR 位址串列中移除其 IP 位址，等到負載降至正常值再恢復其 DNS 設定。如此可使得各伺服器更不易發生故障，進而提升服務的可用性。

## 參考文獻

- [1] Paul Albitz and Cricket Liu, *DNS and BIND, 4th Edition*, O'Reilly, April 2001
- [2] BIND, <http://www.isc.org/BIND/>
- [3] T. Brisco, "DNS Support for Load Balancing," RFC 1794, Rutgers University, April 1995.
- [4] D. M. Dias, William Kish, Rajat Mukherjee, and Renu Tewari, "A scalable and highly available web server," In COMPCON, pages 85–92, 1996.
- [5] Mockapetris, P., "Domain Names - Implementation and Specification", STD 13, RFC 1035, USC/Information Sciences Institute, November 1987.
- [6] Hee Yong Youn, Chansu Yu, Dong Soo Han, and Dongman Lee, "The Approaches for High Available and Fault-Tolerant Cluster Systems," Int'l Workshop on Fault Tolerant Control and Computing (FTCC-1), pp.107-116, May 2000.