# 「Red Hat Certified Architect – Cluster 篇」

## 實戰講座

# 【Single Point Of Failure】

SPOF（Single Point Of Failure）是談 High Availability Cluster 一定會提到的名詞，有人把 SPOF 譯成「單點失效」？從字面上看，不論是中文或英文都讓人有如丈二金鋼，摸不著頭緒；此篇文章便為大家講解 SPOF 及其它相關技術。

# 1 High-availability clusters

Cluster、High Availability Cluster、High performance Computing Cluster 這幾個名詞常讓人混淆。就筆者的看法，「所謂 Cluster 就是由一台以上的機器為了某種特定需求所組成的架構」，根據不同的需求我們可將 Cluster 分為以下三種：

- High availability clusters

增加伺服器和以網路為基礎的應用程式的高可用性及備援性。

- Load Balancing clusters

將服務需求分派給多台伺服器，可視系統負載隨時彈性增加伺服器

- High performance computing clusters

提供同步運算及平行處理的能力

「Red Hat Certified Architect - Cluster 篇」系統文章主要介紹有關「High availability cluster」相關技術。

首先我們先來看維基百科如何定義 High Availability Cluster：

High-availability clusters (also known as HA Clusters or Failover Clusters) are computer clusters that are implemented primarily for the purpose of improving the availability of services which the cluster provides.

什麼是「High Availability」，通常 High Availability 都是譯成「高可用性」，所謂高可用性可想成提高伺服器的提供服務的時間，我們知道伺服器總難免會遇到硬體故障或是當機等天災人禍而無法提供服務。

假設某資料庫伺服器一年平均當機的時間為 1 天，那我們可說此資料庫伺服器（資料庫服務）的可用性可用下列公司粗略計算：

時間區段內系統正常運作時間/時間區段 → （時間區段－當機時間）/時間區段 → 1-系統當機機率 → 1-(1/365) = 99.726%。

假設資料庫停止運作 1 天會造成 200 萬的損失，但此台資料庫伺服器價格為 20 萬。所以就有人想出如果再多買一台一模一樣的備用資料庫伺服器，而且當原來資料庫伺服器當機時，這台備用伺服器可以接手（take over）繼續提供服務，那麼資料庫服務的可用性不就提高了嗎？我們用數學來推算一下，兩台機器同時當機機率為 1/365 x 1/365 。所以這套系統的可用性：

1-(1/365 x 1/365) = 99.999 %

多花 20 萬，換來可用性大幅增加，如果換算成當機時間，不到 5 分鐘，如此一來，與當機一天，損失 200 萬相較之下，此投資顯得相當划算，這也是為什麼企業對 High Availability Cluster 解決方案如此感興趣。

# 2 Single Point Of Failure

High Availability 除了要提供 "*failover*"機制，"failover"機制指的是如果一台伺服器停機或故障, 另一台伺服器可以接手( takeover )啟動應用程式。硬體都必需使用具有 Redundancy 的硬體設備，例如 RAID、Dual Power。

「Redundancy」筆者看到有些文章譯成「冗餘、冗、過多、多餘、贅..」真的是很怪，其實 Redundancy 的意義就是「備援」。至於 fault tolerance 是指「容錯」，所謂「容錯」代表當系統能夠回應非預期性的故障時，可在容許狀況及範圍之下仍然可以繼續運作，無須做任何的的切換或轉移。

至於使用 Redundancy 的硬體設備最主要原因是為了避免造成「**SPOF**」( Single Points Of Failure ) 的情形發生。什麼是「**SPOF**」? 所謂「**SPOF**」是指當某個零件故障會造成整個系統無法正當運作,那麼這個零件就是整個系統中的 Single Points Of Failure。

例如圖 1，SAN Switch 很明顯就是整個系統中的 Single Points Of Failure，只要 SAN Switch 故障，則兩台伺服器均無法存取 Storage，整個系統便無法正常運作。如果要避免此問題，此必須再買一個 SAN Switch，如圖 2 所示。

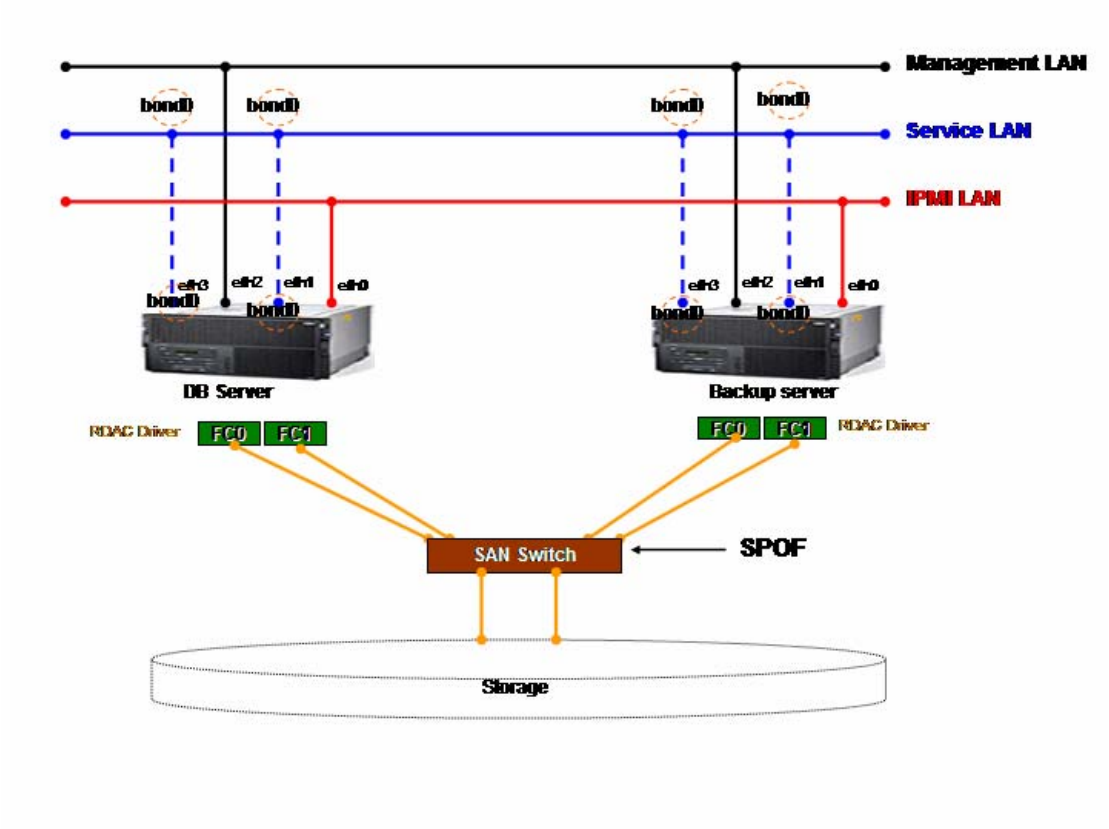


圖 1：SAN Switch 為 SPOF

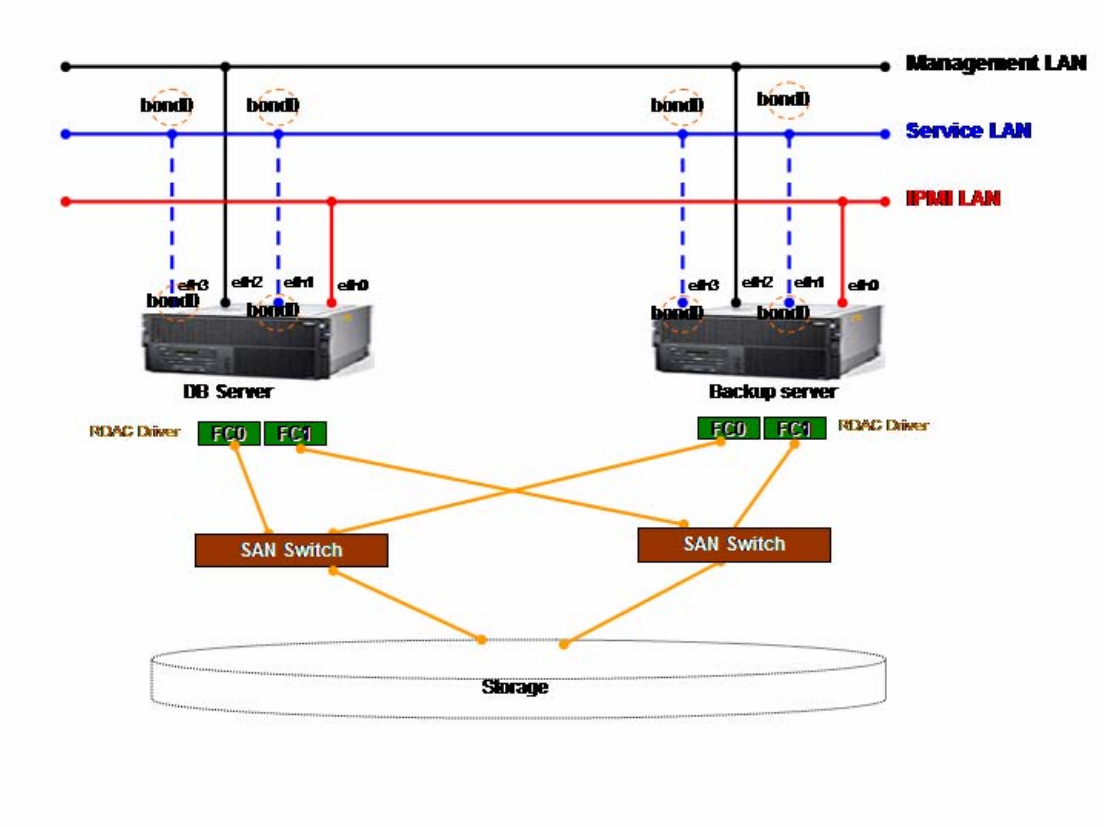


圖 2：SAN Switch 為 Non SPOF

筆者常開玩笑說，要避免 SPOF 的最好方法就是每樣設備，就是「凡是成雙」。雙迴路供電設備、Dual Power 伺服器、Ethernet Switch x 2、SAN Switch x 2、雙控器的儲存設備、Ethernet 兩張、HBA 張兩張...，像圖 3 就是個很標準為了避免 SPOF 所設計的 High Availability Cluster 網路架構。

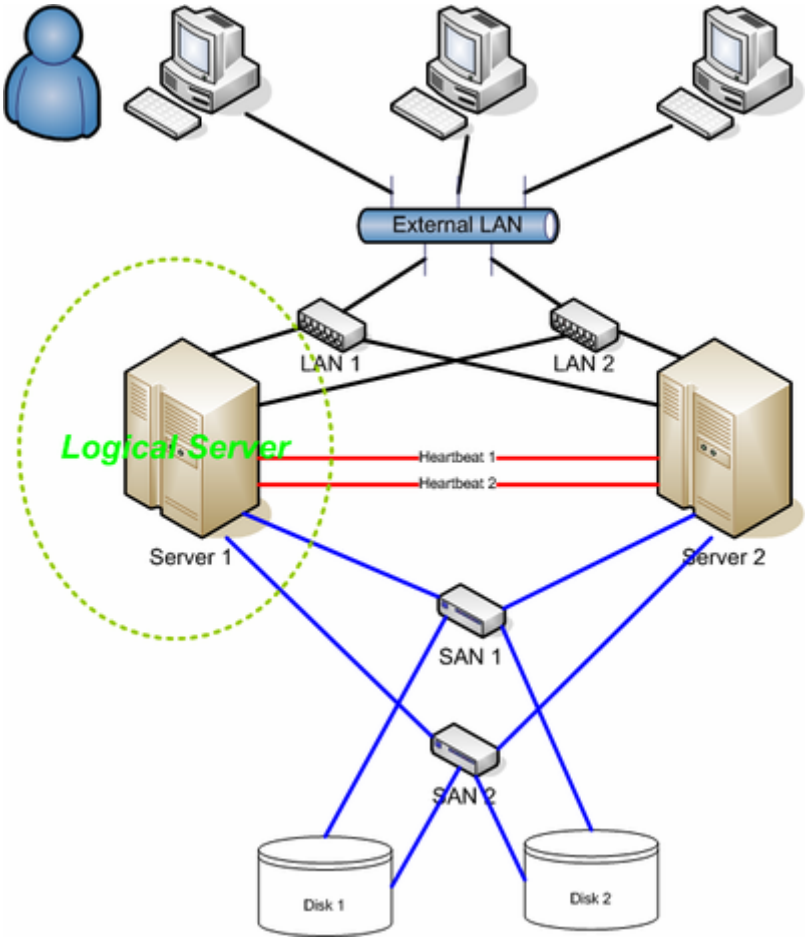


圖 3：2 node High Availability Cluster network diagram

圖片來源：http://en.wikipedia.org/wiki/High-availability_cluster

由圖 3 可發現 server1 及 server2 均配置兩張網路卡及兩張 HBA 卡，這兩張網路卡及 HBA 卡應要有備援的功能，譬如當第一張網卡壞掉或是連接的 Switch 壞掉時，另一張網卡便會 Active，以防止網路中斷。這種網路卡及 HBA 卡備援功能，並不是買了兩張網卡/HBA 卡插上去就好了，除了硬體之外，驅動程式也得做相關設定，下面筆者便是要介紹網卡及 HBA 卡如何設定備援機制。

# 3 Ethernet Bonding

Linux 上所指的 Ethernet bonding 即是將多片網卡虛擬成一張網卡，通常用來達到備援或是分散負載的功能。原本 bonding 的功能是由高階網卡廠商自行開發的驅動程式所提功。但是後來 Linux 提供 bonding 的 module，所以直接利用作業系統所提供 module 便可實作 bondig，無須一定要用廠商所提供的 Driver。

Bonding 之後所有網卡的 IP 和 MAC 將會變成完全相同，然後多出一個 bond0 的虛擬網卡；在 AIX 上，這種機常則稱為 Etherchannel。High Availability Cluster 上的伺服器上對外提供服務的網段，通常會利用兩張網卡在 Active-Backup 的 Ethernet bonding，兩張網卡是同一個 IP，平時只有第一張網卡有作用，如果第一張網卡故障，則另一張網卡便會接手。

作法如下：（假設要將 eth1 及 eth3 虛擬成一張網卡 bond0）

1.vi /etc/sysconfig/network-scripts/ifcfg-eth1

```
DEVICE=eth1
ONBOOT=yes
MASTER=bond0
SLAVE=yes
BOOTPROTO=none
HWADDR=00:1a:64:66:8e:a2
TYPE=Ethernet
```

2.vi /etc/sysconfig/network-scripts/ifcfg-eth3

```
DEVICE=eth3
ONBOOT=yes
MASTER=bond0
SLAVE=yes
BOOTPROTO=none
HWADDR=00:10:18:2E:68:C7
```

3. vi /etc/sysconfig/network-scripts/ifcfg-bond0

```
DEVICE=bond0
ONBOOT=yes
IPADDR=10.3.1.5
NETMASK=255.255.255.0
TYPE=Ethernet
```

4.vi /etc/modprobe.conf

```
alias bond0 bonding
options bond0 miimon=100 mode=1
```

mode=1 代表 active-backup ode，同一時間，只有單一 Slave 網卡運作，若是 Active Slave 網卡失效時則會自動啟用另一張 Slave 網卡。

5.service network restart 重新啟動網路服務即可，並用 ifconfig 檢查網卡狀態（圖 4）。

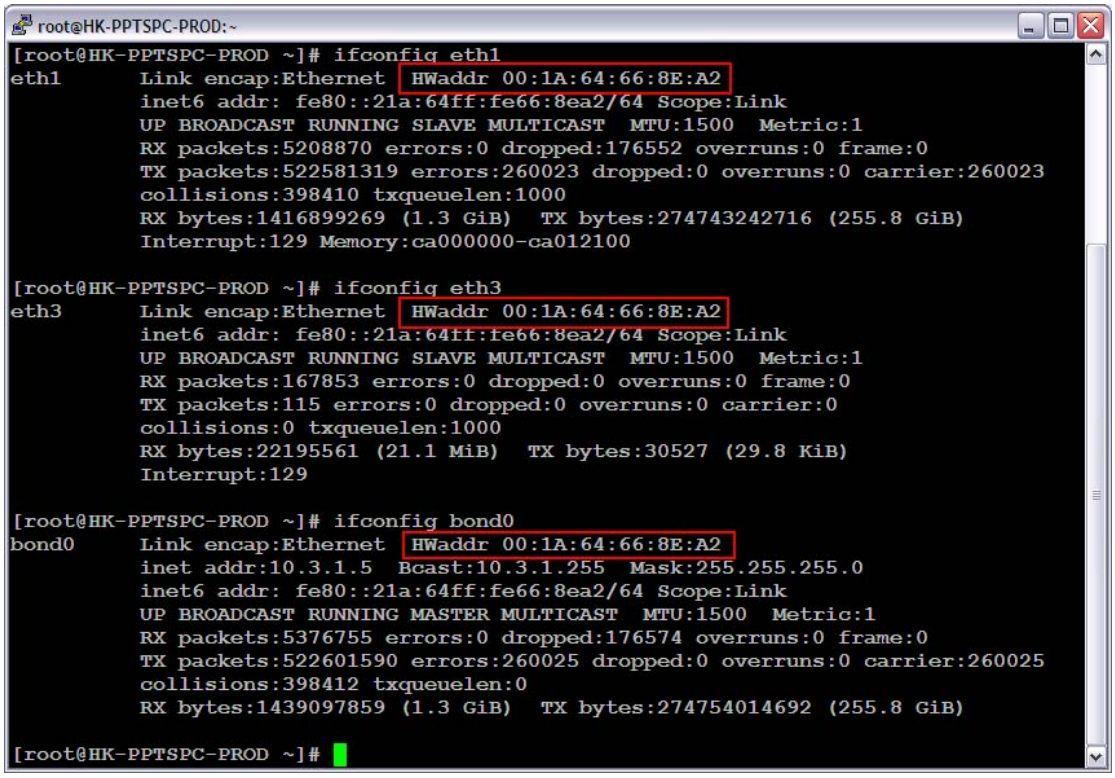


圖 4：Linux Ethernet bonding 畫面

# 4 Fiber multipath

要完成 Firber multipath 功能這個部份就比較麻煩，除了安裝 HBA 卡的 Driver 外，還得依據所連接的儲存設備不同，得選用不同廠商所提供的軟體。例如連接的 EMC 儲存設備就得安裝 EMC PowerPath、如果是 IBM 的 DS4000 系統就得使用 RDAC、如果是 IBM 的 ESS 系統則需安裝 SDD（Subsystem Device Driver）。

而且 IBM RDAC 和 EMC PowerPath 不能並存在同一主機上，兩者會互相衝突，除此之外。還得注意不同的 Linux Kernel 可能得搭配不同版本的 multipath 管理軟體，這個部份，建議讀者一定要搞清楚你們購買的儲存設備所支援的作業系統及 kernel 的版本，而且仔細閱讀該軟體的說明文件（readme.txt）。假設事前沒做好詳細的規劃檢查，可能會遇到作業系統安裝完畢後，找不到搭配此版本作業系統的 multipath 管理軟體。

例如下列網址，就明確寫出如果要連接 IBM 的 ESS, DS6000 和 SVC，作業系統的版本必須為何，Driver 的版本需用那一版本。

http://www-1.ibm.com/support/docview.wss?rs=540&context=ST52G7&dc=D430&uid=ssg1S4000107&loc=en_US&cs=utf-8&lang=en

**表 1、SDD Package for ESS, DS6000, and SAN Volume Controller (SVC)**

| DESCRIPTION | DOCUMENTATION | DOWNLOAD | RELEASE DATE |
|---|---|---|---|
| Platform Red Hat EL 2.1 (x86)<br>SDD v1.6.0.1-11 | SDD 1.6.0.1-11 for Red Hat EL 2.1 (x86) Readme<br>English<br>Byte Size 29165 | SDD 1.6.0.1-11 for Red Hat EL 2.1 (x86)<br>English<br>Byte Size 1362439 | 1/27/06 |
| Platform Red Hat EL 3.0 (ia64)<br>SDD v1.6.0.1-11.2 | SDD 1.6.0.1-11.2 for Red Hat EL 3.0 (ia64) Readme | SDD 1.6.0.1-11.2 for Red Hat EL 3.0 (ia64)<br>English | 5/5/06 |



| | | | |
|---|---|---|---|
| (ESS, DS6000 and DS8000 only). | English<br>Byte Size 29631 | Byte Size 1366514 | |
| Platform SuSE SLES 8.0 (ia64)<br>SDD v1.6.0.1-11.2<br>(ESS, DS6000 and DS8000 only). | SDD 1.6.0.1-11.2 for SuSE SLES 8.0 (ia64) Readme<br>English<br>Byte Size 29631 | SDD 1.6.0.1-11.2 for SuSE SLES 8.0 (ia64)<br>English<br>Byte Size 1292348 | 5/5/06 |
| Platform Red Flag AS & DC 4.1/Asianux 1.0 (x86)<br>SDD v1.6.1.0-4<br>(ESS, DS6000 and DS8000 only). | SDD 1.6.1.0-4 for Asianux 1.0 (x86) Readme<br>English<br>Byte Size 20555 | SDD 1.6.1.0-4 for Asianux 1.0 (x86)<br>English<br>Byte Size 309,790 | 9/11/06 |
| Platform Red Hat EL 3.0 (x86)<br>SDD v1.6.3.0-4 | SDD 1.6.3.0-4 for Red Hat EL 3.0 (x86) Readme<br>English<br>Byte Size 21675 | SDD 1.6.3.0-4 for Red Hat EL 3.0 (x86)<br>English<br>Byte Size 945908 | 10/08/07 |
| Platform Red Hat EL 3.0 (ppc)<br>SDD v1.6.3.0-2 | SDD 1.6.3.0-2 for Red Hat EL 3.0 (ppc) Readme<br>English<br>Byte Size 20853 | SDD 1.6.3.0-2 for Red Hat EL 3.0 (ppc)<br>English<br>Byte Size 622005 | 06/08/07 |
| Platform Red Hat EL 4.0 (x86)<br>SDD v1.6.3.0-5 | SDD 1.6.3.0-5 for Red Hat EL 4.0 (x86) Readme<br>English<br>Byte Size 24448 | SDD 1.6.3.0-5 for Red Hat EL 4.0 (x86)<br>English<br>Byte Size 13413050 | 10/08/07 |
| Platform Red Hat EL 4.0 (ppc)<br>SDD v1.6.3.0-5 | SDD 1.6.3.0-5 for Red Hat EL 4.0 (ppc) Readme<br>English | SDD 1.6.3.0-5 for Red Hat EL 4.0 (ppc)<br>English<br>Byte Size 8653959 | 12/04/07 |

| | | | |
|---|---|---|---|
| | Byte Size 24448 | | |
| Platform Red Hat EL 4.0 (x86_64)<br>SDD v1.6.3.0-5 | SDD 1.6.3.0-5 for Red Hat EL 4.0 (x86_64) Readme<br>English<br>Byte Size 24448 | SDD 1.6.3.0-5 for Red Hat EL 4.0 (x86_64)<br>English<br>Byte Size 6820657 | 10/08/07 |
| Platform SuSE SLES 8.0/ UnitedLinux 1.0 (x86)<br>SDD v1.6.3.0-2 | SDD 1.6.3.0-2 for SuSE SLES 8.0/ UnitedLinux 1.0 (x86) Readme<br>English<br>Byte Size 20853 | SDD 1.6.3.0-2 for SuSE SLES 8.0/ UnitedLinux 1.0 (x86)<br>English<br>Byte Size 663027 | 06/08/07 |
| Platform SuSE SLES 8.0 (ppc)<br>SDD v1.6.3.0-2 | SDD 1.6.3.0-2 for SuSE SLES 8.0 (ppc) Readme<br>English<br>Byte Size 20853 | SDD 1.6.3.0-2 for SuSE SLES 8.0 (ppc)<br>English<br>Byte Size 719550 | 06/08/07 |
| Platform SuSE SLES 9.0 (x86)<br>SDD v1.6.3.0-5 | SDD 1.6.3.0-5 for SuSE SLES 9.0 (x86) Readme<br>English<br>Byte Size 24448 | SDD 1.6.3.0-5 for SuSE SLES 9.0 (x86)<br>English<br>Byte Size 488001 | 12/04/07 |
| Platform SuSE SLES 9.0 (ppc)<br>SDD v1.6.3.0-5 | SDD 1.6.3.0-5 for SuSE SLES 9.0 (ppc) Readme<br>English<br>Byte Size 24448 | SDD 1.6.3.0-5 for SuSE SLES 9.0 (ppc)<br>English<br>Byte Size 393339 | 12/04/07 |
| Platform SuSE SLES 9.0 (x86_64)<br>SDD v1.6.3.0-5 | SDD 1.6.3.0-5 for SuSE SLES 9.0 (x86_64) Readme<br>English<br>Byte Size 24448 | SDD 1.6.3.0-5 for SuSE SLES 9.0 (x86_64)<br>English<br>Byte Size 369568 | 12/04/07 |

# 5 參考資料

Linux High Availability HOWTO

http://www.linux.org.tw/CLDP/OLD/High-Availability-HOWTO/High-Availability-HOWTO.html#toc2

Linux-HA - From Wikipedia, the free encyclopedia

http://en.wikipedia.org/wiki/Linux-HA

Reliability engineering-From Wikipedia, the free encyclopedia

http://en.wikipedia.org/wiki/Reliable_system_design

## 【後記】

還是老話一句，「規劃、規劃、規劃」，在實作RHCS前，就要先想想那些可能會是SPOF，有需要調整架構嗎？應該如何解決？筆者最近接觸Architect的工作，更深切感到事前詳細規劃的重要，如果事前的規劃完善，那麼進入實作階段時，一定事半功倍！

**作者簡介**

林彥明（Alex YM Lin）：現任職於 IBM，負責 HPC 超級電腦、Linux 叢集系統建置、效能調校及技術支援等工作，近來參與 NCHC IBM Cluster 1350（亞洲運算能力僅次日本的超級電腦）及中山大學 p595 HPC 超級電腦專案。具有 RHCA（Red Hat 架構師）、RHCDS（Red Hat Certified Datacenter Specialist）、RHCX（Red Hat 認證主考官）、RHCE、NCLP（Novell Linux 認證專家）、LPIC、IBM AIX …等國際認證。