

Introduction

Linux Bonding Driver 可聚集多個網路埠捆綁(Bonding)成單一個邏輯網路埠，也就是所謂的鏈路聚集(Link Aggregation)的應用。鏈路聚集功能可讓擁有多個 GbE 埠的主機發揮更大效益，得到更可靠或更高效率的網路服務。

一般來說，它提供多種用途的模式(mode)來達到多個網路埠間的 Hot standby 及 Load balance，可預防單點錯誤的發生。預設共有 7 種模式可供選擇，依照使用者不同的負載平衡及容錯的需求來作安裝及設定。

LAB Environment

OS & Version	CentOS 6.3 (2.6.32-279.14.1.el6.x86_64)		
Bonding device	bond0	eth0	Bonding Mode: 0 / balance-rr
		eth1	
	bond1	eth2	Bonding Mode: 6 / balance-alb
		eth3	

Bonding Modes

Mode	Mode Name	Explanation	LB / FT
0	balance-rr	Round-robin policy: 所有的網路埠採循序式 輪流來作封包的傳遞，由第一個網路埠開始傳	Load Balance Fault Tolerance

		送，依序至最後的網路埠，但同一時間只有一個網路埠有在運作。	
1	active-backup	Active & Backup policy: 在構成 Bonding 的群組內，只有一個網路埠是 Active 狀態，其餘都是備援狀態。只有當 Active 的網路埠故障，另外備援的網路埠才會接手轉換成 Active 埠。不需要外部 Switch 額外的設定及支援。	Fault Tolerance
2	balance-xor	XOR policy: 可視為進階版的 Mode 0，不同於 Mode 0 簡單的將封包依序輪流指派給不同的網路埠，Mode 2 加入了 Hash 演算法計算來源端與目的端的 MAC 位址數量來去作網路埠的分派及封包的傳遞。	Load Balance Fault Tolerance
3	broadcast	Broadcast policy: 將所有的封包發送給 Bonding 內所有可用的網路埠作傳遞。	Fault Tolerance
4	802.3ad	IEEE 802.3ad Dynamic link aggregation: 按照 802.3ad 規範，將多個網路埠組成一個共享相同速度及全雙工的鏈路聚集。但 Switch 需要能支援並啟動 802.3ad 協定才能啟用這個模式。以 Cisco 3550 系列為例，首先需將要組成的網	Load Balance Fault Tolerance

		<p>路埠設定為 EtherChannel (或稱為 Trunk Group) , 然後再設定為 LACP 模式 , 以啟動 802.3ad 協定。</p>	
5	balance-tlb	<p>Adaptive transmit load balancing: 這個模式對於封包發送與接收作業採用不同的分配機制。封包發送會依據各個網路埠目前的負載情況(計算相對速度)來作發送的分配; 而封包接收則由目前正在運作中的網路埠來負責, 如果接收埠故障, 則由另一埠套用原先 MAC 位址接手。</p> <p>不需要外部 Switch 額外的設定及支援。</p>	<p>Transmit: Load Balance Fault Tolerance</p> <p>Receive: Fault Tolerance</p>
6	balance-alb	<p>Adaptive load balancing: TLB 的加強版, 採用 ARP 協定, 在接收埠端加入 TLB 沒有的 Load Balance。</p> <p>不需要外部 Switch 額外的設定及支援。</p>	<p>Load Balance Fault Tolerance</p>

Configuration Bonding Devices (Initscripts Support)

1. 修改 `/etc/sysconfig/network-scripts/ifcfg-*` , 指定 Master 及 Slave 的 eth 埠。下列範例將新增兩組 Bonding device , 分別為 Bond0 及 Bond1。Bond0 成員包含 eth0 及

eth1 埠裝置 · Bond1 成員則包含 eth2 及 eth3 。

ifcfg-bond0	ifcfg-eth0	ifcfg-eth1
DEVICE=bond0	DEVICE=eth0	DEVICE=eth1
IPADDR=172.20.144.82	USERCTL=no	USERCTL=no
NETMASK=255.255.255.0	ONBOOT=yes	ONBOOT=yes
GATEWAY=172.20.144.254	MASTER=bond0	MASTER=bond0
ONBOOT=yes	SLAVE=yes	SLAVE=yes
BOOTPROTO=none	BOOTPROTO=none	BOOTPROTO=none
USERCTL=no		
BONDING_OPTS="mode=0 miimon=100"		

ifcfg-bond1	ifcfg-eth2	ifcfg-eth3
DEVICE=bond1	DEVICE=eth2	DEVICE=eth3
IPADDR=1.1.1.2	USERCTL=no	USERCTL=no
NETMASK=255.255.255.0	ONBOOT=yes	ONBOOT=yes
ONBOOT=yes	MASTER=bond1	MASTER=bond1
BOOTPROTO=none	SLAVE=yes	SLAVE=yes
USERCTL=no	BOOTPROTO=none	BOOTPROTO=none

```
BONDING_OPTS="mode=6  
miimon=100"
```

2. 上列 Bonding 選項參數(*BONDING_OPTS*)是直接寫入至各個 ifcfg-bond*設定檔內(註: RHEL 版本需在 v5 之後才可指定 *BONDING_OPTS* 參數)。除了上述方法外,也可將 Bonding 選項參數寫至 /etc/modprobe.conf (註: 通用各個 RHEL 發行版本)。內容參考如下:

```
alias bond0 bonding  
options bond0 -o bond0 mode=0 miimon=100  
  
alias bond1 bonding  
options bond0 -o bond1 mode=6 miimon=100
```

註: 此處若要系統 Bonding 模組載入多個 bond 數量,必須要為每個不同的 bond 命名。例如加入 -o <bond name>。之後,依據上列設定二組 bond0 及 bond1,系統將會參照這二組選項並依序載入兩次。

3. 設定結束後,請重新啟動網路服務。

```
[root@JLAB ~]# service network restart
```

4. 目錄 /proc/net/bonding/ 底下包含了各個 Bonding 裝置的唯讀資訊檔,開啟此檔案可檢視有關 Bonding 的設定、選項及各個網路埠的狀態

I. 檢視 Bond0 狀態

```
[root@JLAB ~]# cat /proc/net/bonding/bond0
```

```
Ethernet Channel Bonding Driver: v3.6.0 (September 26, 2009)
```

```
Bonding Mode: load balancing (round-robin)
```

```
MII Status: up
```

```
MII Polling Interval (ms): 100
```

```
Up Delay (ms): 0
```

```
Down Delay (ms): 0
```

```
Slave Interface: eth1
```

```
MII Status: up
```

```
Speed: 1000 Mbps
```

```
Duplex: full
```

```
Link Failure Count: 0
```

```
Permanent HW addr: 00:50:56:bd:37:7e
```

```
Slave queue ID: 0
```

```
Slave Interface: eth0
```

```
MII Status: up
```

```
Speed: 1000 Mbps
```

```
Duplex: full
```

```
Link Failure Count: 0
```

```
Permanent HW addr: 00:50:56:bd:6a:0e
```

```
Slave queue ID: 0
```

II. 檢視 Bond1 狀態

```
[root@JLAB ~]# cat /proc/net/bonding/bond1
```

```
Ethernet Channel Bonding Driver: v3.6.0 (September 26, 2009)
```

```
Bonding Mode: adaptive load balancing
```

```
Primary Slave: None
```

```
Currently Active Slave: eth2
```

```
MII Status: up
```

```
MII Polling Interval (ms): 100
```

```
Up Delay (ms): 0
Down Delay (ms): 0

Slave Interface: eth2
MII Status: up
Speed: 1000 Mbps
Duplex: full
Link Failure Count: 0
Permanent HW addr: 00:50:56:bd:67:0b
Slave queue ID: 0

Slave Interface: eth3
MII Status: up
Speed: 1000 Mbps
Duplex: full
Link Failure Count: 0
Permanent HW addr: 00:50:56:bd:00:d5
Slave queue ID: 0
```

5. 確認所有網路埠狀態

```
[root@JLAB ~]# ip a
```

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
    master bond0 state UP qlen 1000
    link/ether 00:50:56:bd:6a:0e brd ff:ff:ff:ff:ff:ff
3: eth1: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
    master bond0 state UP qlen 1000
    link/ether 00:50:56:bd:6a:0e brd ff:ff:ff:ff:ff:ff
4: eth2: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
    master bond1 state UP qlen 1000
    link/ether 00:50:56:bd:67:0b brd ff:ff:ff:ff:ff:ff
```

```
5: eth3: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
master bond1 state UP qlen 1000
    link/ether 00:50:56:bd:00:d5 brd ff:ff:ff:ff:ff:ff
6: bond0: <BROADCAST,MULTICAST,MASTER,UP,LOWER_UP> mtu 1500 qdisc
noqueue state UP
    link/ether 00:50:56:bd:6a:0e brd ff:ff:ff:ff:ff:ff
    inet 172.20.144.82/24 brd 172.20.144.255 scope global bond0
    inet6 fe80::250:56ff:febd:6a0e/64 scope link tentative dadfailed
        valid_lft forever preferred_lft forever
7: bond1: <BROADCAST,MULTICAST,MASTER,UP,LOWER_UP> mtu 1500 qdisc
noqueue state UP
    link/ether 00:50:56:bd:67:0b brd ff:ff:ff:ff:ff:ff
    inet 1.1.1.2/24 brd 1.1.1.255 scope global bond1
    inet6 fe80::250:56ff:febd:670b/64 scope link
        valid_lft forever preferred_lft forever
```

6. 當然，也可用工具 `ethtool` 來檢查各個網路埠的狀態。

Reference: <http://www.kernel.org/doc/Documentation/networking/bonding.txt>