Table des matières

Red Hat

On peut utiliser de l'agrégat de cartes réseau pour permettre le redondance. Il faut :

- Modifier le fichier *etc/modules.conf*

```
# Configure modules Bonding
alias bond0 bonding
alias bond1 bonding
options bond0 miimon=100 mode=1 max_bonds=2
options bond1 -o bonding1 miimon=100 mode=1
```

miimon	période de monitoring (en *ms*) des interfaces.
max_bonds	nombre d'interfaces max.
bonding1	indique simplement que *bond1* est la deuxième instance par rapport à *bond0*.

- Modes

mode=0 (balance-rr) Round-robin policy: Transmit packets in sequential order from the first available slave through the last. This mode provides load balancing and fault tolerance.

mode=1 (active-backup) Active-backup policy: Only one slave in the bond is active. A different slave becomes active if, and only if, the active slave fails. The bond's MAC address is externally visible on only one port (network adapter) to avoid confusing the switch. This mode provides fault tolerance. The primary option affects the behavior of this mode.

mode=2 (balance-xor) XOR policy: Transmit based on [(source MAC address XOR'd with destination MAC address) modulo slave count]. This selects the same slave for each destination MAC address. This mode provides load balancing and fault tolerance.

mode=3 (broadcast) Broadcast policy: transmits everything on all slave interfaces. This mode provides fault tolerance.

mode=4 (802.3ad) IEEE 802.3ad Dynamic link aggregation. Creates aggregation groups that share the same speed and duplex settings. Utilizes all slaves in the active aggregator according to the 802.3ad specification.

```
Pre-requisites:
1. Ethtool support in the base drivers for retrieving
the speed and duplex of each slave.
2. A switch that supports IEEE 802.3ad Dynamic link
aggregation.
Most switches will require some type of configuration
to enable 802.3ad mode.
```

mode=5 (balance-tlb) Adaptive transmit load balancing: channel bonding that does not require any special switch support. The outgoing traffic is distributed according to the current load (computed relative to the speed) on each slave. Incoming traffic is received by the current slave. If the receiving slave fails, another slave takes over the MAC address of the failed receiving slave.

```
Prerequisite:
Ethtool support in the base drivers for retrieving the
speed of each slave.
```

mode=6 (balance-alb) Adaptive load balancing: includes balance-tlb plus receive load balancing (rlb) for IPV4 traffic, and does not require any special switch support. The receive load balancing is achieved by ARP negotiation. The bonding driver intercepts the ARP Replies sent by the local system on their way out and overwrites the source hardware address with the unique hardware address of one of the slaves in the bond such that different peers use different hardware addresses for the server.

The most used are the first four mode types…

Also you can use multiple bond interface but for that you must load the bonding module as many as you need. Presuming that you want two bond interface you must configure the /etc/modules.conf as follow:

```
alias bond0 bonding
options bond0 -o bond0 mode=0 miimon=100
alias bond1 bonding
options bond1 -o bond1 mode=1 miimon=100
```

- Créer le(s) fichier(s) de conf *ifcfg-bondX* :

```
DEVICE=bond0
BOOTPROTO=static
IPADDR=XXX.XXX.XXX.XXX
NETMASK=XXX.XXX.XXX.XXX
ONBOOT=yes
TYPE=Ethernet
```

- Créer le(s) fichier(s) de conf *ifcfg-ethX* :

```
DEVICE=eth0
USERCTL=no
ONBOOT=yes
MASTER=bond0
SLAVE=yes
BOOTPROTO=none
```

```
DEVICE=eth1
```

```
USERCTL=no
ONBOOT=yes
MASTER=bond0
SLAVE=yes
BOOTPROTO=none
```

Debian

- Installer le package *ifenslave-2.6*
- Renseigner le fichier */etc/network/interfaces*

```
auto bond0

iface bond0 inet static
    address 88.191.109.11
    netmask 255.255.255.0
    network 88.191.109.0
    broadcast 88.191.109.255
    gateway 88.191.109.1
    slaves eth0 eth1
    bond_mode active-backup
    bond_miimon 100
    bond_downdelay 200
    bond_updelay 200
```

From:
https://unix-bck.ndlp.info/ - **Where there is a shell, there is a way**

Permanent link:
https://unix-bck.ndlp.info/doku.php/informatique:nix:linux:linux_reseau:bonding

Last update: **2011/12/30 15:01**