

Bonding

Ridondanza e load balancing con schede di rete

Matteo Carli
www.matteocarli.com
matteo@matteocarli.com



Chi sono

- Socio e attivista di LUG ACROS e Gentoo Linux.
- Svolgo attività di consulenza presso diverse aziende, principalmente in merito alle tecnologie legate ad Internet, al networking e alla sicurezza informatica.
- Studio “Sicurezza delle reti e dei sistemi informatici” presso l'Università di Milano.



Bonding

E' un driver nato come patch per il Kernel Linux 2.0. Lo sviluppatore originario è Donald Becker uno tra i fondatori del progetto Beowulf.

Negli anni il progetto si è evoluto sviluppando anche ifenslave, un tool userspace per l'amministrazione.



Bonding (2)

- E' volgarmente chiamato anche port-trunking
- In ambienti Microsoft questa tecnica è chiamata teaming.
- E' un driver che permette di aggregare diverse interfacce di rete sotto un'unica interfaccia logica (chiamata master).
- L'interfaccia logica può consentire di suddividere il traffico tra tutte le interfacce.
- L'interfaccia logica può gestire eventuali fault su una delle schede fisiche.



Permette di:

- Aumentare la banda disponibile (throughput) tramite l'uso di algoritmi:
 - Round-Robin
 - XOR logico
 - 802.3ad
 - Load balancing adattivo
- Gestire lo stato del link fisico o logico di tutte le interfacce di rete.
 - Single ARP Monitoring
 - Multile ARP Monitoring
 - MII Monitoring



Modalità disponibili

Esistono 7 modalità diverse di funzionamento del bonding:

- **0 o balance-rr**
 - Load-balancing con Round-Robin.
 - Fornisce fault tolerance.
 - I pacchetti vengono gestiti in maniera sequenziale.
- **1 o active-backup**
 - Fornisce fault tolerance.
 - Una sola interfaccia fisica è attiva.
 - Un solo MAC address utilizzato con l'esterno.



Modalità disponibili (2)

- **2 o balance-xor**
 - Load-balancing con XOR logico.
 - L'algoritmo di XOR usa il MAC sorgente e il MAC destinatario.
 - Fornisce fault tolerance.
- **3 o broadcast**
 - Trasmissione di tutti i pacchetti su tutte le interfacce slave.
 - Fornisce fault tolerance.
- **4 o 802.3ad**
 - Richiede che le interfacce supportino le specifiche 802.3ad.
 - Aggrega tutte le interfacce di slave con le medesime caratteristiche (velocità e duplex).
 - Massimizza la banda passante totale.



Modalità disponibili (3)

- **5 o balance-tlb**

- Fornisce load balancing adattivo sul traffico in uscita.
- Fornisce fault tolerance in entrambe le direzioni.
- L'interfaccia che riceve è la slave attiva.

- **6 o balance-alb**

- Fornisce load balancing adattivo in entrambe le direzioni.
- Il load balancing in entrata viene effettuato tramite protocollo ARP e gestito sequenzialmente con round robin.
- La macchina viene identificata con più MAC dai nodi della rete.
- Utilizza tutte le schede “adattando” il MAC address.



Installazione

- Esempio di installazione di bonding mode 1.
- Sulla colonna di sinistra sarà presente Debian, mentre su quella di destra ci sarà Gentoo.
- Con Debian il kernel usato è il 2.4.27 (necessario un kernel $> 2.4.12$) e su Gentoo è il 2.6.17
- In entrambi i casi ci saranno due interfacce di rete: eth0 e eth1.



Installazione (1)

- make menuconfig
 - Network Device Support
->
 - <M> Bonding driver support
 - apt-get install ifenslave-2.4
 - Editare /etc/network/interfaces con le seguenti specifiche
- make menuconfig
 - Device Drivers ->
 - Networking support
 - <M> Bonding driver support
 - emerge ifenslave
 - Editare /etc/conf.d/net con le seguenti specifiche



Installazione (2)

```
iface bond0 inet static
    address <MIOIP>
    netmask <MIAMASK>
    network <MIOINDRETE>
    gateway <MIOGATEWAY>
    up /sbin/ifenslave bond0 eth0
    up /sbin/ifenslave bond0 eth1
```

Commentare ogni riferimento
alla configurazione di eth0 e di
eth1

```
slaves_bond0="eth0 eth1"
config_bond0=( "<MIOIP> netmask
<MIAMASK> brd
<MIOBROADCAST>" )
routes_bond0=( "default gw
<MIOGATEWAY>" )
```

Commentare ogni riferimento
alla configurazione di eth0 e di
eth1



Installazione (3)

- nano /etc/modutils/arch/i386
- Aggiungere:
 - alias bond0 bonding
 - options bond0 mode=1 miimon=100
- Update-modules
- reboot
- rc-update del net.eth0 default
- rc-update del net.eth1 default
- In -sf /etc/init.d/net.lo /etc/init.d/net.bond0
- rc-update add net.bond0 default
- nano /etc/modules.autoload.d/kernel-2.6
- Aggiungere:
 - bonding mode=1 miimon=100
- reboot



Installazione (4): test

ifconfig

bond0 Link encap:Ethernet HWaddr **00:40:D4:CB:E3:1C**
inet addr:<MIOIP> Bcast:<MIOBROADCAST> Mask:255.255.255.0
UP BROADCAST RUNNING **MASTER** MULTICAST MTU:1500 Metric:1
RX packets:12643 errors:0 dropped:0 overruns:0 frame:0
TX packets:20759 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:2374753 (2.2 Mb) TX bytes:3427302 (3.2 Mb)

eth0 Link encap:Ethernet HWaddr **00:40:D4:CB:E3:1C**
UP BROADCAST RUNNING **SLAVE** MULTICAST MTU:1500 Metric:1
RX packets:12294 errors:0 dropped:0 overruns:0 frame:0
TX packets:20688 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:2347175 (2.2 Mb) TX bytes:3422364 (3.2 Mb)
Interrupt:16 Base address:0x2000

eth1 Link encap:Ethernet HWaddr **00:40:D4:CB:E3:1C**
UP BROADCAST **NOARP SLAVE** MULTICAST MTU:1500 Metric:1
RX packets:349 errors:0 dropped:0 overruns:0 frame:0
TX packets:73 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:27578 (26.9 Kb) TX bytes:5406 (5.2 Kb)
Interrupt:17 Base address:0x4000

interfaccia di loopback omessa per questione di spazio



Installazione (5): test

```
# cat /proc/net/bonding/bond0  
Ethernet Channel Bonding Driver: v2.6.5 (November 4, 2005)
```

Bonding Mode: fault-tolerance (active-backup)

Primary Slave: None

Currently Active Slave: eth0

MII Status: up

MII Polling Interval (ms): 100

Up Delay (ms): 0

Down Delay (ms): 0

Slave Interface: eth0

MII Status: up

Link Failure Count: 1

Permanent HW addr: **00:40:D4:CB:E3:1C**

Slave Interface: eth1

MII Status: up

Link Failure Count: 1

Permanent HW addr: **00:40:B4:C0:A4:B3**



Installazione (6): considerazioni

- L'interfaccia di bond è settata come MASTER, eth0 come slave primaria e eth1 come slave non utilizzata
- Se staccassimo il collegamento sulla eth0, la eth1 diventerebbe l'interfaccia slave primaria
- Si notino i MAC address delle tre interfacce.
- Controlliamo che nelle route della macchina non siano ancora configurate eth0 o eth1.



Approfondimenti

- Gestione del link monitoring.
- Configurazioni degli switch in base al modo di bonding utilizzato.
- Schede di rete supportate.
- Pianificazione del fault tolerance



Gestione del link monitoring (ARP)

- Vengono inviate query ARP ad uno o più nodi sulla rete
- Se arriva un ARP “replies” il link è considerato up altrimenti down.
- Passaggio parametri al modulo:
 - ARP verso un singolo nodo:
 - › arp_interval=60 arp_ip_target=192.168.0.110
 - ARP verso host multipli:
 - › arp_interval=60
arp_ip_target=192.168.0.110,192.168.0.111,192.168.0.112



Gestione del link monitoring (MII)

- Viene controllato il link fisico (elettrico) sull'interfaccia di rete.
- Tramite chiamate in kernel-space il driver ottiene lo stato del link
 - Specificando `use_carrier=1` (default) il modulo interrogherà il driver utilizzando `netif_carrier`
 - Specificando `use_carrier=0` il modulo interrogherà l'interfaccia tramite una `ioctl` sul registro MII.



Configurazioni degli switch in base al modo di bonding utilizzato

- Con i modi che seguono non vi è bisogno di configurazione (salvo casi particolari):
 - Active-backup
 - Balance-tlb
 - Balance-alb
- Con i modi che seguono vi è bisogno di configurazioni e/o di supporto a specifiche particolari:
 - 802.3ad
 - Balance-rr
 - balance-xor



Schede di rete supportate

- Utilizzando le schede più conosciute non si hanno problemi:
 - Realtek RTL-8139/8139C/8139C
 - Via Vt6102
 - Realtek RTL-8139D
 - Intel E100/E1000
- Ho notato problemi con alcune schede Gigabit (anche Realtek).

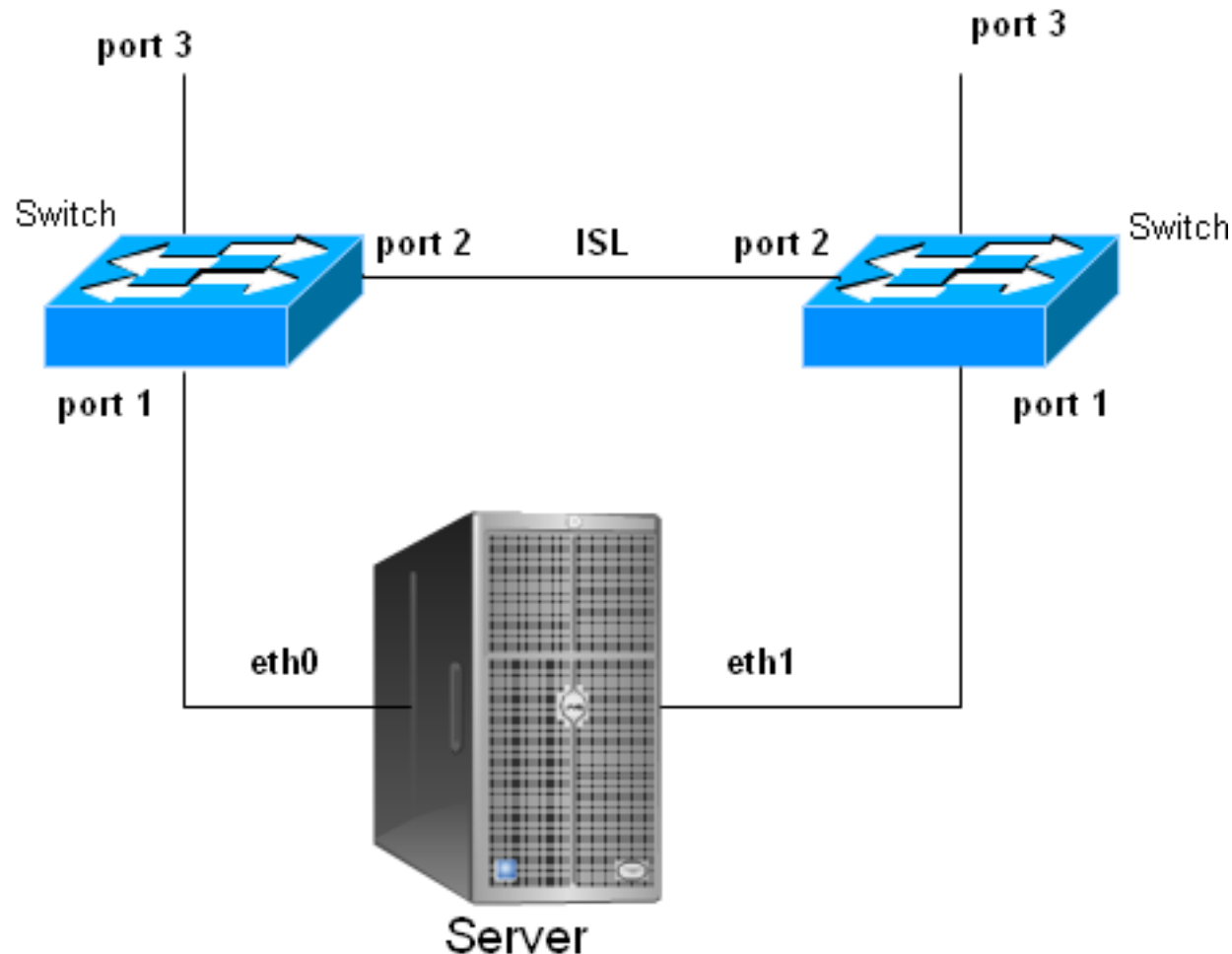


Pianificazione del fault tolerance

- Non basta fare bonding, attaccare i cavi e andare a prendere un caffè.
- Il fault tolerance fatto con l'opzione MII è il meno invasivo ma anche il meno efficace.
- Connettere entrambe le interfacce di rete ad un medesimo switch (o host) non esclude il caso sia lo switch ad avere un fault.



Pianificazione del fault tolerance (2)



In questo esempio si aumenta la ridondanza collegando, ogni interfaccia del server, ad uno switch diverso. Su una configurazione di rete di questo tipo sono consigliati i modi: active-backup e broadcast in modo che i due collegamenti siano indipendenti in caso di fault.



Documentazione

- [/usr/src/linux/Documentation/networking/bonding.txt](#)
- [sourceforge.net/projects/bonding](#)
- [www.cisco.com/en/US/tech/tk389/tk390/technologies_tech_note09186a0080094665.shtml](#) (protocollo ISL)
- [www.beowulf.org](#)



Domande?

?

