Configuración de NIC Bonding y Link Aggregation

Aitor Lázaro Sánchez

Contenido

Máquinas y SO

* Europa e Io: 16 cores y 32 GB de RAM
* Ganimedes y Calisto: 24 cores y 64 GB de RAM
* Dos HDD de 300GB a 15000 rpm en RAID 0
* SO utilizado: Debian Squeeze de 64 bits.
* Particiones: 1GB para /boot, 2GB para swap y 544 para /

Red

- Objetivo:
  - Bonding mediante tres NIC's para lograr 3 Gbps en la comunicación entre las cuatro máquinas.
  - Bonding en modo 802.3ad
  - Link Aggregation en el switch mediante LACP
- Logrado:
  - Bonding en modo 0 que funciona a una velocidad que ronda desde los 2'15 hasta los 2'3 Gbps
  - Switch desconfigurado.

NIC Bonding

* Creación de una interfaz de red virtual que controla a varias físicas.

* Los objetivos del NIC Bonding son el aumento de ancho de banda y la tolerancia a fallos de las interfaces.

Modos de NIC Bonding en Linux

* Modo 0 o balance rr: Usa el alogoritmo Round Robin y es el modo por defecto. Ofrece balanceo de carga y tolerancia a fallos.
* Modo 1 o Active backup: Su funcionamiento consiste en tener activa una interfaz y el resto como backup. Ofrece tolerancia a fallos.
* Modo 2 o Balance XOR: Todo lo que vaya para una MAC en concreto se envía por la misma interfaz. Ofrece balanceo y tolerancia a fallos.
* Modo 3 o Broadcast: Envía todo por todas las interfaces. Ofrece tolerancia a fallos.

Modos de NIC Bonding en Linux

* Modo 4 o 802.3ad: Estandar. Permite agregar varios enlaces para aumentar el ancho de banda. Todos los interfaces deben ser de la misma velocidad y el Switch compatible con el protocolo. Ofrece balanceo y tolerancia a fallos.
* Modo 5 o Balance-TLB: Balancea el envío según la carga de las interfaces. Necesita compatibilidad con ethtool. Ofrece balanceo y tolerancia.
* Modo 6 o Balance-ALB: Similar al modo 5 pero con el añadido de balancear la recepción. Tolerancia a fallos y balanceo.

Link Aggregation

* El objetivo de esta técnica es la configuración de varios puertos para que virtualmente actúen como uno solo.

* Con esto se logra la redundancia de conexión para evitar fallos y la mejora del ancho de banda.

Configuración del NIC Bonding en Debian Squeeze

* Configuración final:
  * Bonding en modo 0
  * Interfaces usadas: eth1, eth2, eth3
* Para la configuración del bonding en Debian es necesaria la instalación del paquete ifenslave-2.6.

```
aptitude install ifenslave-2.6
```

Parametros de ifenslave-2.6

* **arp_interval:** Indica cada cuántos milisegundos se envía un ARP reply. Por defecto = 0.
* **arp_ip_target:** Indica cual será la IP destino. Hasta 16 destinos.
* **downdelay:** Especifica los milisegundos que tardará en bajar la interfaz cuando se detecte un error. El valor por defecto es 0.
* **updelay:** Indica cuantos milisegundos se tardará en activar una tarjeta de red esclava cuando se detecte un error en una interfaz. El valor por defecto es 0

Parametros de ifenslave-2.6

* **max_bonds:** Indica cuantas interfaces virtuales se crearán al iniciar el móduclo. Por defecto es 1.
* **miimon:** Indica cada cuantos milisegundos se va a comprobar el estado de los enlaces, por defecto es 0.
* **mode:** Indica el modo de bonding.
* **primary:** Solo se utiliza en el modo 1 y sirve para indicar que interfaz va a estar como activa y cual o cuales como esclavas.

Parametros de ifenslave-2.6

* **use_carrier:** Su valor puede ser 0 o 1, si es 1 indica que usará una llamada netif_carrier_ok() del módulo de la tarjeta de red para la detección por MII.
* **xmit_hash_policy:** Especifica la politica de transmisión para los modos 2 y 4, pueden ser:
  * **Layer 2:** Similar al modo 2: todo lo que vaya para una IP lo mandará por la misma interfaz.
  * **Layer 3+4:** Usa niveles superiores al de enlace y actúa según el tipo de tráfico. No es completamente compatible con el 802.3ad
* **slaves:** Indica que interfaces actúan como esclavas en el bonding.

Configuración de bond1

* Lo primero es bajar todas las interfaces que funcionarán como esclavas:

  ifdown ethx

* Modificamos el fichero /etc/network/interfaces y añadimos las siguientes lineas:

```
iface bond1 inet static
 address 192.168.222.13
 netmask 255.255.255.0
 bond-slaves eth1 eth2 eth3
 bond-mode 0
 mtu 9000
 bond-miimon 100
```

Y reiniciamos networking

/etc/init.d/networking restart

Configuración de bond1

* Otra forma de configurarlo: Mediante comandos.
* Activamos el módulo indicandole el modo de bonding y el valor de miimon:

  modprobe bonding mode=0 (o el nombre) miimon=100

* Levantamos bond1:

  ifconfig bond1 192.168.222.13 netmask 255.255.255.0 mtu 9000 up

* Añadimos los esclavos:

  ifenslave bond1 eth1 eth2 eth3

Configuración de bond1

* Para bajar y desconfigurar el bond hay que realizar los siguientes pasos:
* Bajar las interfaces

  ifdown eth1 eth2 eth3
* Bajar el bond1

  ifconfig bond1 down
* Quitar el módulo

  modprobe –r bonding

Comprobación de la configuración

* Con ifconfig:

```
bond1     Link encap:Ethernet  HWaddr 00:25:90:69:f8:a9
          inet addr:192.168.222.13  Bcast:192.168.222.255  Mask:255.255.255.0
          inet6 addr: fe80::225:90ff:fe69:f8a9/64 Scope:Link
          UP BROADCAST RUNNING MASTER MULTICAST  MTU:9000  Metric:1
          RX packets:1847300 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1428607 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1732704907 (1.6 GiB)  TX bytes:6066067471 (5.6 GiB)


eth1      Link encap:Ethernet  HWaddr 00:25:90:69:f8:a9
          UP BROADCAST RUNNING SLAVE MULTICAST  MTU:9000  Metric:1
          RX packets:796589 errors:0 dropped:0 overruns:0 frame:0
          TX packets:476431 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1412769379 (1.3 GiB)  TX bytes:2023694953 (1.8 GiB)
          Memory:fe7e0000-fe800000
```

Comprobación de la configuración

* Para comprobar la correcta configuración del bonding podemos echar mano de los siguientes ficheros:

  /proc/net/bonding/bond1

  /sys/class/net/bond1/bonding/mode

* Otros ficheros donde se muestran los parámetros están en:

  /sys/class/net/bond1

  /sys/class/net/bond1/bonding/

Link Aggregation en el Switch

* Tras muchos intentos no se logró configurar el Link Aggregation en el switch SMC8028L2 ni en modo LACP ni en modo estático.

* Se decidió dejarlo sin configurar y en modo 0 ya que así se lograba el aumento de ancho de banda.

Pruebas de rendimiento

* Mediante iperf se han hecho pruebas con 50 clientes de forma simultanea, estos son los resultados de algunas pruebas:
  * Io como servidor, Ganimedes como cliente:

    [SUM] 0.0-10.0 sec 2.51 GBytes 2.16 Gbits/sec
  * Ganimedes como servidor, Calisto como cliente:

    [SUM] 0.0-10.6 sec 2.66 GBytes 2.27 Gbits/sec
  * Calisto como servidor, Io como cliente:

    [SUM] 0.0-10.4 sec 2.64 GBytes 2.18 Gbits/sec

Intentos de configuración

* Lograr que funcione ha sido más complicado de lo esperado.
* Se han probado todos los tipos de bonding.
* Los modos 1 y 3 son los que primero se obviaron.
* Los modos 2, 5 y 6 tampoco mostraban mejora alguna
* El modo 4, seguramente por error a la hora de configurar el switch, no dio el resultado deseado.
* Especial mención a la casi nula (por no decir nula) documentación acerca de bonding con Gigabit Ethernet, que es algo bastante común.

Intentos de configuración

* Diferencia entre Ubuntu y Debian Wheezy:
  * En la configuración de la interfaz:

    ```
    auto eth1
    iface eth1 inet manual
    bond-master bond1
    ```

  * En la configuración del bond:

    ```
    bond-slaves none
    ```

  * En Debian no hace falta la configuración de la interfaz, con indicar los slaves en la configuración del bond ya vale.

    ```
    bond-slaves eth1 eth2 eth3
    ```

Intentos de configuración en modo 802.3ad

* A la hora de intentar configurar el bond en modo 4 (802.3ad) los siguientes parámetros eran prácticamente invariables:

```
auto bond1
iface bond0 inet static
 address 192.168.222.13
 netmask 255.255.255.0
 bond-mode 4
 bond-miimon 100
 bond-slaves none

auto eth1
iface eth1 inet manual
bond-master bond0
```

Intentos de configuración en modo 802.3ad

* Desde un principio se añadieron las siguientes líneas que fueron eliminadas por no ser necesarias:

  bond downdelay 200
  bond-updelay 200

* Otro parametro, al final eliminado debido a que se ha usado el modo 0, pero que era correcto es el siguiente:

  bond-lacp-rate 1

Intentos de configuración en modo 802.3ad

* Otra línea que también estuvo, pero que se quitó debido a que solo es compatible con el modo 1 es la siguiente:

  bond-primary eth1 eth2 eth3

* Por último se añadió al final, tal y como ponía en alguna página de internet, la siguiente línea, que al final se quedó en la configuración final:

  mtu 9000

Intentos de configuración en modo 802.3ad

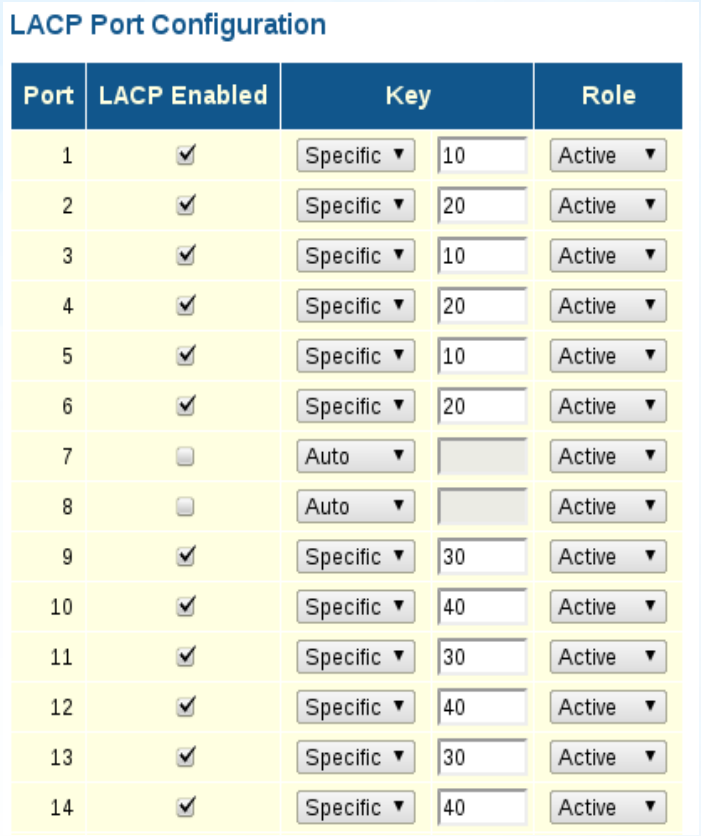
* Última configuración probada en modo 4:

```
auto bond1
iface bond0 inet static
 address 192.168.222.13
 netmask 255.255.255.0
 bond-mode 4
 bond-miimon 100
 bond-slaves eth1 eth2 eth3
 bond-lacp-rate 1
 mtu 9000
```

Intentos de configuración en modo 802.3ad

* En el switch:

Importante desconectar los puertos a la hora de modificar la configuración y reiniciar el switch tras las modificaciones



LACP Port Configuration

Port	LACP Enabled	Key		Role
1	☑	Specific	10	Active
2	☑	Specific	20	Active
3	☑	Specific	10	Active
4	☑	Specific	20	Active
5	☑	Specific	10	Active
6	☑	Specific	20	Active
7	☐	Auto		Active
8	☐	Auto		Active
9	☑	Specific	30	Active
10	☑	Specific	40	Active
11	☑	Specific	30	Active
12	☑	Specific	40	Active
13	☑	Specific	30	Active
14	☑	Specific	40	Active

Intentos de configuración en modo 802.3ad

* Una vez configurados los nodos y el switch NO pasaba de los 980Mbps (sus mu…)
* Para asegurar que no era un problema de que no cogiesen la configuración se comprobaron los siguientes ficheros:

  /sys/class/net/bond1/bonding/ad_partner_key

  /sys/class/net/bond1/bonding/ad_actor_key

Intentos de configuración

* Se realizaron otros intentos en otros modos:
  * El modo 1 se descartó ya que solo funciona una interfaz a la vez.
  * El modo 3, Broadcast, al enviar la misma información por todos los nodos a la vez no solo no aumentaba sino que diminuía a unos 400Mbps.
  * Los modos 2 tampoco era muy aconsejable por el balanceo mediante la MAC destino.
  * Los modos 5 y 6 no mejoraron el rendimiento.

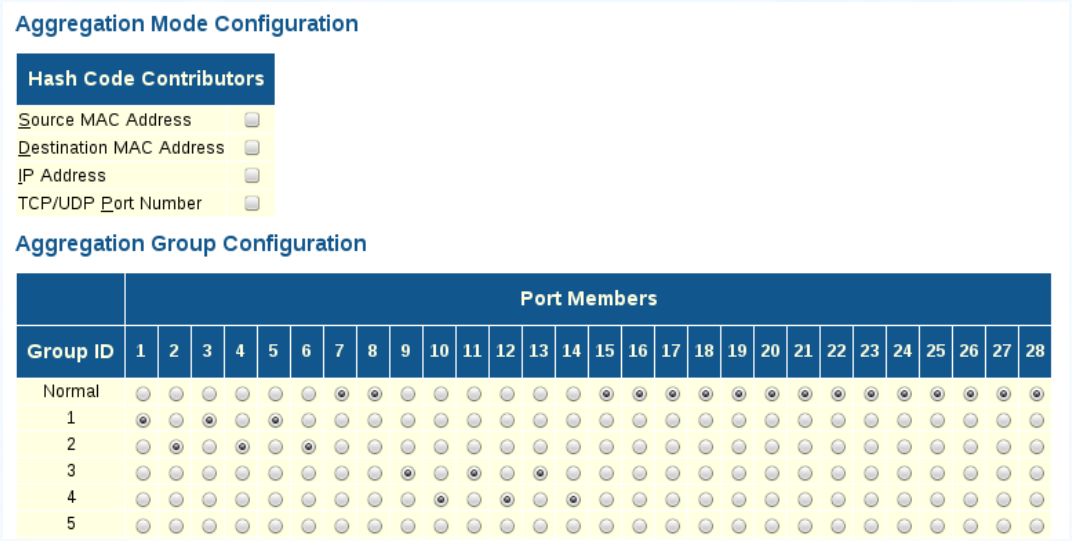
Intentos de configuración

* No hubo ningún cambio importante a la hora de cambiar los modos:

```
iface bond1 inet static
address 192.168.222.13
netmask 255.255.255.0
bond-slaves eth1 eth2 eth3
bond-mode <número de modo>
mtu 9000
bond-miimon 100
```

Intentos de configuración

* Durante estas pruebas en otros modos el switch se configuró tanto en LACP como en estático.
* Configuración en estático: Ningún hash.



Aggregation Mode Configuration

Hash Code Contributors	
Source MAC Address	☐
Destination MAC Address	☐
IP Address	☐
TCP/UDP Port Number	☐

Aggregation Group Configuration

	Port Members																										
Group ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Normal	○	○	○	○	○	○	◉	◉	○	○	○	○	○	○	◉	◉	◉	◉	◉	◉	◉	◉	◉	◉	◉	◉	◉	◉
1	◉	○	◉	○	◉	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
2	○	◉	○	◉	○	◉	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
3	○	○	○	○	○	○	○	○	◉	○	◉	○	◉	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
4	○	○	○	○	○	○	○	○	○	◉	○	◉	○	◉	○	○	○	○	○	○	○	○	○	○	○	○	○	○
5	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

Intentos de configuración

* Seguía sin mejorar el rendimiento.
* Aitor se empieza a plantear la opción de pegarse un tiro.
* Jesús encuentra la página por la que se obra el milagro. Alberto ejecuta.
* Se realizan las siguientes modificaciones:
  * Se vuelve al modo 0.
  * Se desconfigura el Switch completamente.

```
echo "3000" > /proc/sys/net/core/netdev_max_backlog
ifconfig bond1 txqueuelen 10000
ethtool -G eth1 rx 3072 tx 3072
```

* ¡FUNCIONÓ!

Intentos de configuración

* Caso para Iker Jimenez:
* Cómo se hicieron varios cambios a la vez hay que hacer pruebas para documentar que provocó el cambio.
* Se desconfigura todo (máquinas y switch) y se reincian tanto los nodos como el switch.
* Se prueba sin añadir los cambios finales…
* Y FUNCIONA
* Posiblemente el error fuese una mala configuración guardada en el switch…

* Fin.