

bonding原理

Linux bonding 驱动程序提供一种聚合方法，将多个网络接口转换或者虚拟成单个逻辑上的接口。bonding驱动程序提供了多种绑定模式。在目前的 Linux 众多发行版本中，bonding 程序已成为一个内核模块。可以使用modinfo命令查看bonding内核模块信息。通常情况下，网卡只能接收目标MAC地址是本网卡的MAC的帧，而拒绝目标其他MAC地址的帧数据。bonding的实现是通过让网卡工作在 promisc 模式下，从而可以接收网络中的所有帧数据。同时两个物理网卡将被修改为相同的MAC地址，当网卡收到数据时，将数据交给 bonding 驱动处理。将多个网卡绑定聚合在一起，形成一个逻辑上的网卡接口，在用户看来就是一个网卡。利用 bonding 实现了网卡硬件层面的冗余。

bonding模式

Linux有7种网卡绑定模式。

1. Mode=0 负载均衡模式 (balance -rr)

依次传输数据包，数据包依次从两个网卡经过。此模式提供负载均衡和容错能力。

2. Mode=1 主备模式 (Action-backup)

只有一个设备处于活动状态，当一个设备发生故障时，马上由另一个设备接替工作Bonding的MAC地址是唯一的，当发生主备切换时，MAC地址也会切换。

3. Mode=3 广播模式 (broadcast)

在所有接口上传送所有报文，提供容错能力，此模式不需要交换的特殊处理。

详细绑定模式参考：[深度分析Linux下双网卡绑定七种模式](#)

bonding配置

配置

在 `/etc/sysconfig/network-scripts` 目录下创建bond设备文件，以 `ifcfg-bondN` 的形式命名。

```
1 # /etc/sysconfig/network-scripts/ifcfg-bond0 配置项
2 TYPE="Ethernet"
3 NAME="bond0"
4 DEVICE="bond0"
5 ONBOOT="yes"
6 BONDING_OPTS="miimon=100 mode=1"
7 BOOTPROTO="none"
8 IPADDR=172.18.26.10
9 PREFIX=16
10 GATEWAY=172.18.0.1
11 DNS1=172.18.0.1
12 DNS2=144.144.144.144
13 DOMAIN=xuejinwei.com
```

说明： `BONDING_OPTS="miimon=100 mode=1"` 配置项中，**miimon** 是用来进行链路监测的。如果 `miimon=100`，那么系统每100ms监测一次链路连接状态，如果有一条线路不通就转入另一条线路。`mode=1` 表示使用mode1模式。

修改 `/etc/sysconfig/network-scripts/ifcfg-eth0` 和 `/etc/sysconfig/network-scripts/ifcfg-eth1` 即两块网卡的配置文件，分别加入：

```
1 # /etc/sysconfig/network-scripts/ifcfg-eth0
2 MASTER=bond0
3 SLAVE=yes
```

```
1 # /etc/sysconfig/network-scripts/ifcfg-eth1
2 MASTER=bond0
3 SLAVE=yes
```

说明： MASTER=bond0 表示将此设备加入的bond名称；

SLAVE=yes 表示此设备隶属于bond设备。当把网卡加入bond设备后，此配置文件的所有配置项失效。

启动

配置文件修改后，重新启动网络服务。

```
1 | systemctl restart network
```

说明：在CentOS上，有network和NetworkManager两个服务管理网卡，为了避免冲突，需要将NetworkManager关闭。

查看

使用 `ip addr show` 命令查看网卡设备信息，发现bond0设备已经分配到设置的IP地址。eth0和eth1的MAC地址变为一样。

```
[ root@node0 ~ ]# ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    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 00:0c:29:8d:66:f6 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 00:0c:29:8d:66:f6 brd ff:ff:ff:ff:ff:ff
4: bond0: <BROADCAST,MULTICAST,MASTER,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP qlen 1000
    link/ether 00:0c:29:8d:66:f6 brd ff:ff:ff:ff:ff:ff
    inet 172.18.26.10/16 brd 172.18.255.255 scope global bond0
        valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe8d:66f6/64 scope link
        valid_lft forever preferred_lft forever
```

bond设备的状态信息保存在 `/proc/net/bonding/` 目录下，两块设备中，当前正在工作的网卡的eth0。

```
[ root@node0 ~ ]# cat /proc/net/bonding/bond0
Ethernet Channel Bonding Driver: v3.7.1 (April 27, 2011)

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
Speed: 1000 Mbps
Duplex: full
Link Failure Count: 0
Permanent HW addr: 00:0c:29:8d:66:f6
Slave queue ID: 0

Slave Interface: eth1
MII Status: up
Speed: 1000 Mbps
Duplex: full
Link Failure Count: 0
Permanent HW addr: 00:0c:29:8d:66:00
Slave queue ID: 0
```

物理上断开一块网卡的连接，检测bond是否生效。

```
[ root@node0 ~ ]# cat /proc/net/bonding/bond0
Ethernet Channel Bonding Driver: v3.7.1 (April 27, 2011)

Bonding Mode: fault-tolerance (active-backup)
Primary Slave: None
Currently Active Slave: eth1
MII Status: up
MII Polling Interval (ms): 100
Up Delay (ms): 0
Down Delay (ms): 0

Slave Interface: eth0
MII Status: down
Speed: Unknown
Duplex: Unknown
Link Failure Count: 1
Permanent HW addr: 00:0c:29:8d:66:f6
Slave queue ID: 0

Slave Interface: eth1
MII Status: up
Speed: 1000 Mbps
Duplex: full
Link Failure Count: 2
Permanent HW addr: 00:0c:29:8d:66:00
Slave queue ID: 0
```

断开eth0网卡，eth1成为Active 网卡。

删除bonding

```
1 # 删除bonding配置文件
2 rm -f ifcfg-bond0
3 # 卸载binding内核模块
4 modprobe -r bonding
5 # 重新启动网络服务
6 systemctl restart network
```

使用nmcli工具配置网卡绑定

使用nmcli添加bond链接生成配置文件

```
1 [ root@node0 ~ ]# nmcli connection add type bond con-
   name bond0 ifname bond0 mode active-backup
2 Connection 'bond0' (c410edba-565e-424d-af6e-
   854b00efd637) successfully added.
```

给bond设备添加地址网关等网卡基本信息

```
1 [ root@node0 ~ ]# nmcli connection modify bond0
   ipv4.addresses 172.18.26.100/16
   connection.autoconnect yes ipv4.dns-search
   xuejinwei.com ipv4.dns 172.18.0.1 ipv4.method manual
   ipv4.gateway 172.18.0.1
```

将网卡eth0和eth1添加到bond中

```
1 [ root@node0 ~ ]# nmcli connection add con-name
   bond0-eth0 type bond-slave ifname eth0 master bond0
2 Connection 'bond0-eth0' (be1e34fd-2daa-4d18-81dd-
   2ac0a2beb039) successfully added.
3 [ root@node0 ~ ]# nmcli connection add con-name
   bond0-eth1 type bond-slave ifname eth1 master bond0
4 Connection 'bond0-eth1' (6e347543-5515-4871-902e-
   6bf94ec8a6b7) successfully added.
```

查看网卡链接

```
1 [ root@node0 ~ ]# nmcli connection show
2 NAME          UUID
3 TYPE          DEVICE
4 bond0         c410edba-565e-424d-af6e-854b00efd637
   bond         bond0
5 eth0          dd6f01ea-0e88-45e9-86a6-b51a5cac12e9
   802-3-ethernet eth0
6 eth1          c849e694-c20e-3a10-a591-d0cfccac3361
   802-3-ethernet eth1
7 bond0-eth0    be1e34fd-2daa-4d18-81dd-2ac0a2beb039
   802-3-ethernet --
8 bond0-eth1    6e347543-5515-4871-902e-6bf94ec8a6b7
   802-3-ethernet --
```

启用 网卡的bind的配置文件，即将生成的配置文件关联到设备

```
1 [ root@node0 ~ ]# nmcli connection up bond0-eth1
2 Connection successfully activated (D-Bus active
   path:/org/freedesktop/NetworkManager/ActiveConnection
   /5)
3 [ root@node0 ~ ]# nmcli connection up bond0-eth0
4 Connection successfully activated (D-Bus active
   path:/org/freedesktop/NetworkManager/ActiveConnection
   /5)
```

启用bond设备

```
1 [ root@node0 ~ ]# nmcli connection up bond0
```

停止bond

停止bond设备，删除bond链接文件，并重启network服务

```
1 [ root@node0 ~ ]# nmcli connection down bond0
2 [ root@node0 ~ ]# nmcli connection delete bond0-ens33
3 [ root@node0 ~ ]# nmcli connection delete bond0-ens37
```

