

# 网口聚合自动化工具

## 使用指导手册



## 版权声明与使用须知

### 版权声明

©2023 浙江宇视科技有限公司。保留一切权利。

未经浙江宇视科技有限公司（下称“本公司”）书面许可，任何单位和个人不得擅自摘抄、复制本手册内容的部分或全部，并不得以任何形式传播。

本手册描述的产品中可能包含本公司及可能存在的许可人享有版权的软件。未经相关权利人许可，任何人不得以任何形式对前述软件进行包括但不限于复制、分发、修改、摘录、反编译、反汇编、解密、反向工程、出租、转让、分许可等侵犯软件著作权的行为。

### 使用须知

由于产品版本升级等原因，本手册内容会不定期更新。

本手册仅作为使用指导，其内所有陈述、信息和建议等均不构成任何明示或暗示的担保。本手册中的图形、图表或照片等仅用于说明示例，可能与实际产品存在差异，请以实物为准。

## 一. 工具概况

端口聚合（又称为链路聚合），将交换机上的多个端口在物理上连接起来，在逻辑上捆绑在一起，形成一个拥有较大宽带的端口，可以实现负载分担，从而增强带宽，提供冗余。使用端口聚合技术，当一条链路出现故障时，另一条链路会继续工作。

获取途径：联系平台二线获取

查看网络端口信息：ifconfig

网卡配置文件目录：/etc/sysconfig/network-scripts

注：网口聚合的七种模式如下。

第一种模式：mode=0，即：(balance-rr) Round-robin policy（平衡轮循环策略）默认，有高可用（容错）和负载均衡的功能

特点：传输数据包顺序是依次传输（即：第 1 个包走 eth0，下一个包就走 eth1...一直循环下去，直到最后一个传输完毕），此模式提供负载平衡和容错能力；但是我们知道如果一个连接或者会话的数据包从不同的接口发出的话，中途再经过不同的链路，在客户端很有可能会出现数据包无序到达的问题，而无序到达的数据包需要重新要求被发送，这样网络的吞吐量就会下降。

第二种模式：mode=1，即：(active-backup) Active-backup policy（主-备份策略）只有高可用（容错）功能

特点：只有一个设备处于活动状态，当一个宕掉另一个马上由备份转换为主设备。mac 地址是外部可见得，从外面看来，bond 的 MAC 地址是唯一的，以避免 switch(交换机)发生混乱。此模式只提供了容错能力；由此可见此算法的优点是可以提供高网络连接的可用性，但是它的资源利用率较低，只有一个接口处于工作状态，在有 N 个网络接口的情况下，资

源利用率为  $1/N$

第三种模式：mode=2，即：(balance-xor) XOR policy（平衡策略） 不常用

特点：基于指定的传输 HASH 策略传输数据包。缺省的策略是：(源 MAC 地址 XOR 目标 MAC 地址) % slave 数量。其他的传输策略可以通过 xmit\_hash\_policy 选项指定，此模式提供负载平衡和容错能力

第四种模式：mode=3，即：broadcast（广播策略） 不常用

特点：在每个 slave 接口上传输每个数据包，此模式提供了容错能力

第五种模式：mode=4，即：(802.3ad) IEEE 802.3ad Dynamic link aggregation（IEEE 802.3ad 动态链接聚合）

特点：创建一个聚合组，它们共享同样的速率和双工设定。根据 802.3ad 规范将多个 slave 工作在同一个激活的聚合体下。

外出流量的 slave 选举是基于传输 hash 策略，该策略可以通过 xmit\_hash\_policy 选项从缺省的 XOR 策略改变到其他策略。需要注意的是，并不是所有的传输策略都是 802.3ad 适应的，尤其考虑到在 802.3ad 标准 43.2.4 章节提及的包乱序问题。不同的实现可能会有不同的适应性。

必要条件：

条件 1：ethtool 支持获取每个 slave 的速率和双工设定

条件 2：switch(交换机)支持 IEEE 802.3ad Dynamic link aggregation

条件 3：大多数 switch(交换机)需要经过特定配置才能支持 802.3ad 模式

第六种模式：mode=5，即：(balance-tlb) Adaptive transmit load balancing（适配器传输负载均衡） 不常用

特点：不需要任何特别的 switch(交换机)支持的通道 bonding。在每个 slave 上根据当前的负载（根据速度计算）分配外出流量。如果正在接受数据的 slave 出故障了，另一个 slave 接管失败的 slave 的 MAC 地址。

必要条件：

ethtool 支持获取每个 slave 的速率

第七种模式：mode=6，即：(balance-alb) Adaptive load balancing（适配器适应性负载均衡） 有高可用（容错）和负载均衡的功能

特点：该模式包含了 balance-tlb 模式，同时加上针对 IPV4 流量的接收负载均衡(receive load balance, rlb)，而且不需要任何 switch(交换机)的支持。接收负载均衡是通过 ARP 协商实现的。bonding 驱动截获本机发送的 ARP 应答，并把源硬件地址改写为 bond 中某个 slave 的唯一硬件地址，从而使得不同的对端使用不同的硬件地址进行通信。

来自服务器端的接收流量也会被均衡。当本机发送 ARP 请求时，bonding 驱动把对端的 IP 信息从 ARP 包中复制并保存下来。当 ARP 应答从对端到达时，bonding 驱动把它的硬件地址提取出来，并发起一个 ARP 应答给 bond 中的某个 slave。使用 ARP 协商进行负载均衡的一个问题是：每次广播 ARP 请求时都会使用 bond 的硬件地址，因此对端学习到这个硬件地址后，接收流量将会全部流向当前的 slave。这个问题可以通过给所有的对端发送更新（ARP 应答）来解决，应答中包含他们独一无二的硬件地址，从而导致流量重新分布。

当新的 slave 加入到 bond 中时，或者某个未激活的 slave 重新激活时，接收流量也要重

新分布。接收的负载被顺序地分布（round robin）在 bond 中最高速的 slave 上

当某个链路被重新接上，或者一个新的 slave 加入到 bond 中，接收流量在所有当前激活的 slave 中全部重新分配，通过使用指定的 MAC 地址给每个 client 发起 ARP 应答。下面介绍的 updelay 参数必须被设置为某个大于等于 switch(交换机)转发延时的值，从而保证发往对端的 ARP 应答 不会被 switch(交换机)阻截。

必要条件：

条件 1: ethtool 支持获取每个 slave 的速率；

条件 2：底层驱动支持设置某个设备的硬件地址，从而使得总是有个 slave(curr\_active\_slave)使用 bond 的硬件地址，同时保证每个 bond 中的 slave 都有一个唯一的硬件地址。如果 curr\_active\_slave 出故障，它的硬件地址将会被新选出来的 curr\_active\_slave 接管

其实 mode=6 与 mode=0 的区别：mode=6，先把 eth0 流量占满，再占 eth1，...ethX；而 mode=0 的话，会发现 2 个口的流量都很稳定，基本一样的带宽。而 mode=6，会发现第一个口流量很高，第 2 个口只占了小部分流量

## 二．工具使用方法

上传并运行 bond\_unv231127.sh 脚本

1. 输入需要创建的网卡名称，建议为 bond0：

```

[root@node1 home]# sh bond_unv231127.sh
欢迎使用网卡绑定脚本
该脚本需在root用户下完成,开始效验用户
当前执行用户校验正确!
您将需要输入以下信息完成配置
-----
1-新创建的聚合网卡绑定名称, 建议为bond0 (数字0)
2-网卡绑定模式,XOR方式聚合设为2,主备模式设为1,静态负载均衡设为0
3-服务器IP地址,
4-服务器子网掩码位数,如255.255.255.0,与PREFIX=24表示意思相同
5-服务器网关,
6-需要绑定的实际网卡名称
-----
1-新创建网卡绑定名称:

```

2. 选择网卡聚合模式, 具体参考上面七种聚合模式的资料 (默认 type 2 balance-xor):

```

2-请选择网卡绑定模式: 0 | 1 | 2 | 3 | 4 | 5 | 6 :
(Default: type 2):

```

3. 输入服务器聚合的 IP 地址:

```

您已经选择绑定模式为2 balance-xor
3-服务器IP地址:

```

4. 输入子网掩码位数 (0-32):

```

4-服务器子网掩码位数:16
您已经输入设备子网掩码位数为: 16

```

5. 输入服务器网关:

```

5-服务器网关:206.206.0.1
您已经输入设备网关为: 206.206.0.1

```

6. 输入需要绑定的网卡名称,多个网卡之间用英文空格隔开 (服务器不同, 网卡名称也不同, 根据服务器实际情况选择对应网卡)

```

以下是系统识别到的所有网卡:
enp26s0f0
enp1s0
enp26s0f1
enp2s0
enp26s0f2
enp3s0
enp26s0f3
6-请输入需要绑定的网卡名称,多个网卡之间用英文空格隔开
网卡名称:

```

```

6-请输入需要绑定的网卡名称,多个网卡之间用英文空格隔开
网卡名称:enp26s0f0 enp26s0f1

```

```

网卡名称:enp26s0f0 enp26s0f1
连接 "bond0" (564b903d-70f6-49c3-b3fe-6d4fa615a1a8) 已成功添加。
连接 "bond-slave-enp26s0f0" (af8d6228-49de-420f-b829-19bfca36a116) 已成功添加。
连接 "bond-slave-enp26s0f1" (55055ed0-dcb3-48a3-aa7f-536359d9dad5) 已成功添加。
连接已成功激活 (master waiting for slaves) (D-Bus 活动路径: /org/freedesktop/NetworkManager/ActiveConnection/35)
[root@node1 home]#

```

聚合完成, 查看 ip 是否生效。

```

连接已成功 (master waiting for slaves) (bond3s 启动路径: /org/freedesktop/networkmanager/ActiveConnection/33)
[root@node1 home]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    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: enp26s0f0: <NO-CARRIER,BROADCAST,MULTICAST,SLAVE,UP> mtu 1500 qdisc mq master bond0 state DOWN group default qlen 1000
    link/ether 00:00:00:00:03:14 brd ff:ff:ff:ff:ff:ff
3: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:a0:c9:00:00:00 brd ff:ff:ff:ff:ff:ff
4: enp26s0f1: <NO-CARRIER,BROADCAST,MULTICAST,SLAVE,UP> mtu 1500 qdisc mq master bond0 state DOWN group default qlen 1000
    link/ether 00:00:00:00:03:14 brd ff:ff:ff:ff:ff:ff permaddr 00:00:00:00:03:15
5: enp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:a0:c9:00:00:31 brd ff:ff:ff:ff:ff:ff
6: enp26s0f2: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state DOWN group default qlen 1000
    link/ether 00:00:00:00:03:16 brd ff:ff:ff:ff:ff:ff
7: enp3s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:7f:89:29:67:b6 brd ff:ff:ff:ff:ff:ff permaddr 00:a0:c9:00:00:00
    inet 206.206.31.35/16 brd 206.206.255.255 scope global noprefixroute enp3s0
        valid_lft forever preferred_lft forever
    inet6 fe80::27f:89ff:fe29:67b6/64 scope link
        valid_lft forever preferred_lft forever
8: enp26s0f3: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state DOWN group default qlen 1000
    link/ether 00:00:00:00:03:17 brd ff:ff:ff:ff:ff:ff
15: bond0: <NO-CARRIER,BROADCAST,MULTICAST,MASTER,UP> mtu 1500 qdisc noqueue state DOWN group default qlen 1000
    link/ether 00:00:00:00:03:14 brd ff:ff:ff:ff:ff:ff
    inet 206.206.31.36/16 brd 206.206.255.255 scope global noprefixroute bond0
        valid_lft forever preferred_lft forever

```

如果之前已聚合网口，想要重新聚合，脚本会识别聚合网口，提示是否删除重新聚合

```

valid_lft forever preferred_lft forever
[root@node1 home]# sh bond_unv231127.sh
欢迎使用网卡绑定脚本
该脚本需在root用户下完成,开始效验用户
当前执行用户校验正确!
当前服务器已创建网口聚合, 请选择是否删除重建创建[yes/no]:

```

输入 yes，将删除聚合网口，重新配置。

```

当前执行用户校验正确!
当前服务器已创建网口聚合, 请选择是否删除重建创建[yes/no]:
yes
正在删除聚合网口
成功删除连接 "bond0" (564b903d-70f6-49c3-b3fe-6d4fa615a1a8)。
成功删除连接 "bond-slave-enp26s0f0" (af8d6228-49de-420f-b829-19bfca36a116)。
成功删除连接 "bond-slave-enp26s0f1" (55055ed0-dcb3-48a3-aa7f-536359d9dad5)。
聚合网口删除成功!
您将需要输入以下信息完成配置
-----
1- 新创建的聚合网卡绑定名称, 建议为bond0 (数字0)
2- 网卡绑定模式, XOR方式聚合设为2, 主备模式设为1, 静态负载均衡设为0
3- 服务器IP地址,
4- 服务器子网掩码位数, 如255.255.255.0, 与PREFIX=24表示意思相同
5- 服务器网关,
6- 需要绑定的实际网卡名称

```

输入 no，退出

```

no
程序退出

```

此脚本仅修改服务器，交换机根据实际需求配置。





---

视无界 智以恒