深度分析Linux下双网卡绑定七种模式

2013年11月7日　　10:39

linuxinterfacedelay负载均衡网络算法

现在一般的企业都会使用双网卡接入，这样既能添加网络带宽，同时又能做相应的冗余，可以说是好处多多。而一般企业都会使用linux操作系统下自带的网卡绑定模式，当然现在网卡产商也会出一些针对windows操作系统网卡管理软件来做网卡绑定（windows操作系统没有网卡绑定功能 需要第三方支持）。进入正题，linux有七种网卡绑定模式：0. round robin，1.active-backup，2.load balancing (xor)， 3.fault-tolerance (broadcast)， 4.lacp， 5.transmit load balancing， 6.adaptive load balancing。

第一种：bond0:round robin

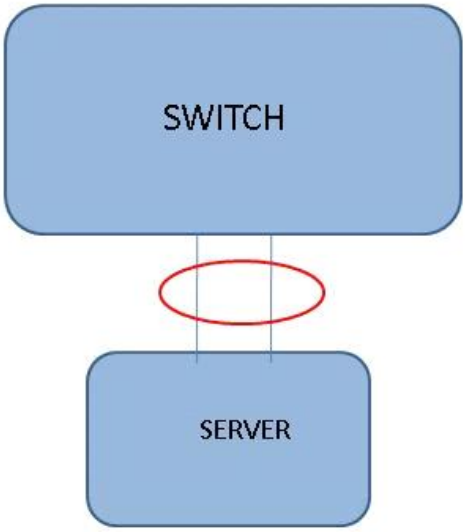
标准：round-robin policy: Transmit packets in sequential order from the first available slave through the last. This mode provides load balancing and fault tolerance.

特点：（1）所有链路处于负载均衡状态，轮询方式往每条链路发送报文，基于per packet方式发送。服务上ping 一个相同地址：1.1.1.1 双网卡的两个网卡都有流量发出。负载到两条链路上，说明是基于per packet方式 ，进行轮询发送。（2）这模式的特点增加了带宽，同时支持容错能力，当有链路出问题，会把流量切换到正常的链路上。

实际绑定结果：

cat /proc/net/bonding/bond0
Ethernet Channel Bonding Driver: v3.6.0 (September 26, 2009)
Bonding Mode: load balancing (round-robin)　　－－－－－ＲＲ的模式
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: 0
Permanent HW addr: 74:ea:3a:6a:54:e3
Slave Interface: eth1
MII Status: up
Link Failure Count: 0

应用拓扑：交换机端需要配置聚合口，cisco叫port channel。拓扑图如下：



第二种：bond1:active-backup

标准文档定义：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.

模式的特点：一个端口处于主状态 ，一个处于从状态，所有流量都在主链路上处理，从不会有任何流量。当主端口down掉时，从端口接手主状态。

实际绑定结果：

root@1:~# cat /proc/net/bonding/bond0
Ethernet Channel Bonding Driver: v3.6.0 (September 26, 2009)
Bonding Mode: fault-tolerance (active-backup) ——backup模式
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: 0
Permanent HW addr: 74:ea:3a:6a:54:e3
Slave Interface: eth1
MII Status: up
Link Failure Count: 0
Permanent HW addr: d8:5d:4c:71:f9:94

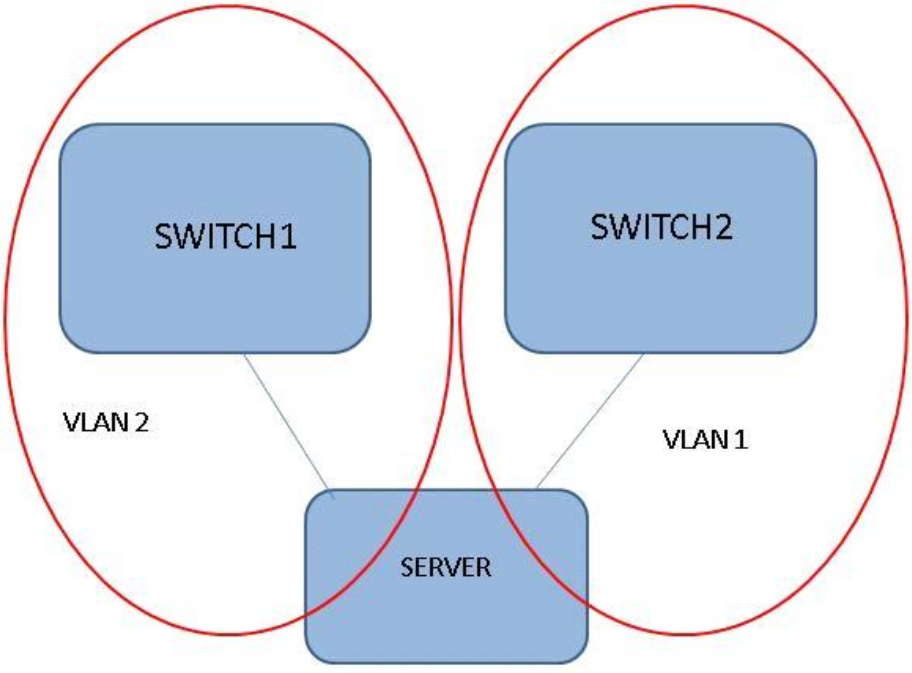
应用拓扑：这种模式接入不需要交换机端支持，随便怎么接入都行。

第三种：bond2:load balancing (xor)

标准文档描述：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.

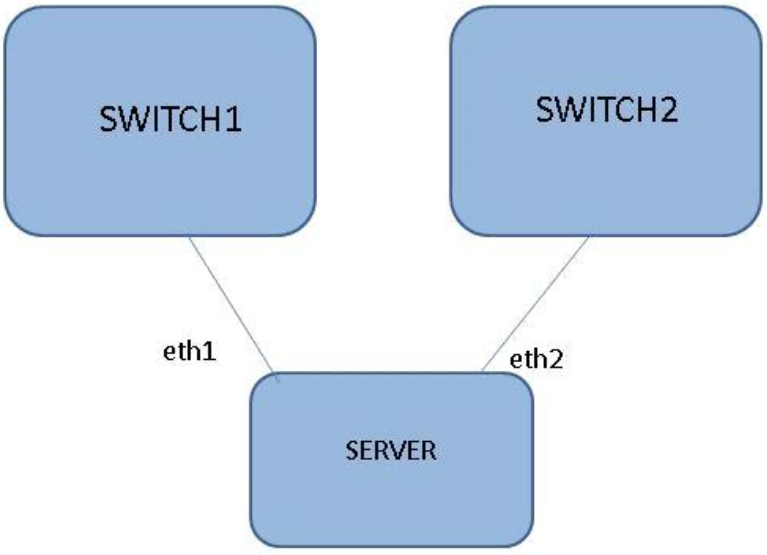
特点：该模式将限定流量，以保证到达特定对端的流量总是从同一个接口上发出。既然目的地是通过MAC地址来决定的，因此该模式在“本地”网络配置下可以工作得很好。如果所有流量是通过单个路由器（比如 “网关”型网络配置，只有一个网关时，源和目标mac都固定了，那么这个算法算出的线路就一直是同一条，那么这种模式就没

有多少意义了。），那该模式就不是最好的选择。和balance-rr一样，交换机端口需要能配置为“port channel”。这模式是通过源和目标mac做hash因子来做xor算法来选路的。
实际绑定结果：
[root@localhost ~]# cat /proc/net/bonding/bond0
Ethernet Channel Bonding Driver: v3.0.3 (March 23, 2006)
Bonding Mode: load balancing (xor) ——配置为xor模式
Transmit Hash Policy: layer2 (0)
MII Status: up
MII Polling Interval (ms): 100
Up Delay (ms): 0
Down Delay (ms): 0
Slave Interface: eth1
MII Status: up
Link Failure Count: 0
Permanent HW addr: 00:d0:f8:40:f1:a0
Slave Interface: eth2
MII Status: up
Link Failure Count: 0
Permanent HW addr: 00:d0:f8:00:0c:0c
应用拓扑：同bond0一样的应用模型。这个模式也需要交换机配置聚合口。
第四种：bond3:fault-tolerance (broadcast)
标准文档定义：Broadcast policy: transmits everything on all slave interfaces. This mode provides fault tolerance.
特点:这种模式的特点是一个报文会复制两份往bond下的两个接口分别发送出去,当有对端交换机失效，我们感觉不到任何downtime,但此法过于浪费资源;不过这种模式有很好的容错机制。此模式适用于金融行业，因为他们需要高可靠性的网络，不允许出现任何问题。
实际绑定结果：
root@ubuntu12:~/ram# cat /proc/net/bonding/bond0
Ethernet Channel Bonding Driver: v3.6.0 (September 26, 2009)
Bonding Mode: fault-tolerance (broadcast) ——- fault-tolerance 模式
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: 0
Permanent HW addr: 74:ea:3a:6a:54:e3
Slave Interface: eth1
MII Status: up
Link Failure Count: 0
Permanent HW addr: d8:5d:4c:71:f9:94
应用拓扑：如下：



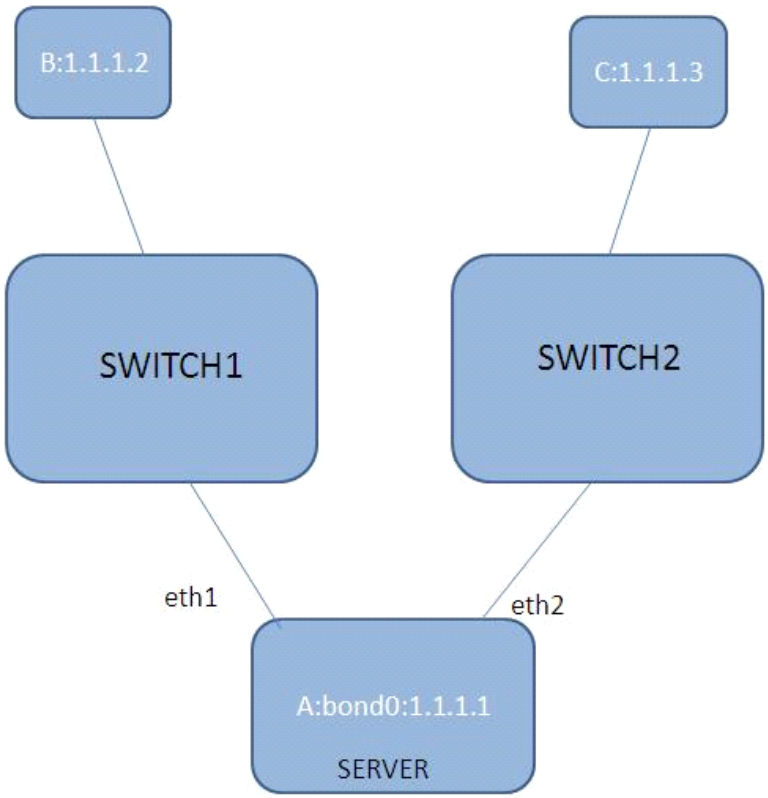
这种模式适用于如下拓扑，两个接口分别接入两台交换机，并且属于不同的vlan，当一边的网络出现故障不会影响服务器另一边接入的网络正常工作。而且故障过程是0丢包。下面展示了这种模式下ping信息：
64 bytes from 1.1.1.1: icmp_seq=901 ttl=64 time=0.205 ms
64 bytes from 1.1.1.1: icmp_seq=901 ttl=64 time=0.213 ms (DUP!) —dup为重复报文
64 bytes from 1.1.1.1: icmp_seq=902 ttl=64 time=0.245 ms
64 bytes from 1.1.1.1: icmp_seq=902 ttl=64 time=0.254 ms (DUP!)
64 bytes from 1.1.1.1: icmp_seq=903 ttl=64 time=0.216 ms
64 bytes from 1.1.1.1: icmp_seq=903 ttl=64 time=0.226 ms (DUP!)
从这个ping信息可以看到，这种模式的特点是，同一个报文服务器会复制两份分别往两条线路发送，导致回复两份重复报文，这种模式有浪费资源的嫌疑。
第五种：bond4:lacp
标准文档定义：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.
特点：802.3ad模式是IEEE标准，因此所有实现了802.3ad的对端都可以很好的互操作。802.3ad 协议包括聚合的自动配置，因此只需要很少的对交换机的手动配置（要指出的是，只有某些设备才能使用802.3ad）。802.3ad标准也要求帧按顺序（一定程度上）传递，因此通常单个连接不会看到包的乱序。802.3ad也有些缺点：标准要求所有设备在聚合操作时，要在同样的速率和双工模式，而且，和除了balance-rr模式外的其它bonding负载均衡模式一样，任何连接都不能使用多于一个接口的带宽。
此外，linux bonding的802.3ad实现通过对端来分发流量（通过MAC地址的XOR值），因此在“网关”型配置下，所有外出（Outgoing）流量将使用同一个设备。进入（Incoming）的流量也可能在同一个设备上终止，这依赖于对端802.3ad实现里的均衡策略。在“本地”型配置下，路两将通过 bond里的设备进行分发。

实际绑定结果：
root@:~# cat /proc/net/bonding/bond0
Ethernet Channel Bonding Driver: v3.6.0 (September 26, 2009)
Bonding Mode: IEEE 802.3ad Dynamic link aggregation
Transmit Hash Policy: layer2 (0)
MII Status: up
MII Polling Interval (ms): 100
Up Delay (ms): 0
Down Delay (ms): 0
802.3ad info
LACP rate: slow
Aggregator selection policy (ad_select): stable
Active Aggregator Info:
Aggregator ID: 1
Number of ports: 1
Actor Key: 9
Partner Key: 1
Partner Mac Address: 00:00:00:00:00:00
Slave Interface: eth0
MII Status: up
Link Failure Count: 0
Permanent HW addr: 74:ea:3a:6a:54:e3
Aggregator ID: 1
Slave Interface: eth1
MII Status: up
Link Failure Count: 0
Permanent HW addr: d8:5d:4c:71:f9:94
Aggregator ID: 2
应用拓扑：应用拓扑同bond0,和bond2一样，不过这种模式除了配置port channel之外还要在port channel聚合口下开启LACP功能，成功协商后，两端可以正常通信。否则不能使用。
交换机端配置：
interface AggregatePort 1 配置聚合口
interface GigabitEthernet 0/23
port-group 1 mode active 接口下开启lacp 主动模式
interface GigabitEthernet 0/24
port-group 1 mode active
第六种：bond5: transmit load balancing
标准文档定义：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.
特点：balance-tlb模式通过对端均衡外出（outgoing）流量。既然它是根据MAC地址进行均衡，在“网关”型配置（如上文所述）下，该模式会通过单个设备来发送所有流量，然而，在“本地”型网络配置下，该模式以相对智能的方式（不是balance-xor或802.3ad模式里提及的XOR方式）来均衡多个本地网络对端，因此那些数字不幸的MAC地址（比如XOR得到同样值）不会聚集到同一个接口上。
不像802.3ad，该模式的接口可以有不同的速率，而且不需要特别的交换机配置。不利的一面在于，该模式下所有进入的（incoming）流量会到达同一个接口；该模式要求slave接口的网络设备驱动有某种ethtool支持；而且ARP监控不可用。
实际配置结果：
cat /proc/net/bonding/bond0
Ethernet Channel Bonding Driver: v3.0.3 (March 23, 2006)
Bonding Mode: transmit load balancing ——TLB模式
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: eth1
MII Status: up
Link Failure Count: 0
Permanent HW addr: 00:d0:f8:40:f1:a0
Slave Interface: eth2
MII Status: up
Link Failure Count: 0
Permanent HW addr: 00:d0:f8:00:0c:0c
应用拓扑：这个模式下bond成员使用各自的mac，而不是上面几种模式是使用bond0接口的mac。



如上图，设备开始时会发送免费arp，以主端口eth1的mac为源，当客户端收到这个arp时就会在arp缓存中记录下这个mac对的ip。而在这个模式下，服务器每个端口在ping操作时，会根据算法算出出口，地址不断变化时他，这时会负载到不同端口。实验中ping1.1.1.3时往eth2发送，源mac为00:D0:F8:00:0C:0C，ping1.1.1.4是往eth1发送，

源mac为00:D0:F8:40:F1:A0，以此类推，所以从服务器出去的流量负载到两条线路，但是由于服务发arp时只用00:D0:F8:40:F1:A0，这样客户端缓冲记录的是00:D0:F8:40:F1:A0对的ip，封装时目标mac：00:D0:F8:40:F1:A0。这样进入服务的流量都只往eth1（00:D0:F8:40:F1:A0）走。设备会一直发入snap报文，eth1发送源为00d0.f840.f1a0的snap报文，eth2发送源为00d0.f800.0c0c的snap报文。这个snap报文mac和目标mac一样都是网卡本地mac，源ip和目标ip也一样，这个报文的作用是检测线路是否正常的回环报文。
注：可以通过修改bond0的mac地址来引导他发修改后的源mac的免费arp（MACADDR=00:D0:F8:00:0C:0C）
第七种：bond6:adaptive load balancing
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。所有端口都会收到对端的arp请求报文，回复arp回时，bond驱动模块会截获所发的arp回复报文，根据算法算到相应端口，这时会把arp回复报文的源mac，send源mac都改成相应端口mac。从抓包情况分析回复报文是第一个从端口1发，第二个从端口2发。以此类推。
(还有一个点：每个端口除发送本端口回复的报文，也同样会发送其他端口回复的报文，mac还是其他端口的mac)这样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(这个算法和上面一样，比如算到1口，就给发送arp请求，1回复时mac用1的mac)。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。这个问题通过给所有的对端发送更新（ARP应答）来解决，往所有端口发送应答,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。
实际配置结果：
root@:/tmp# cat /proc/net/bonding/bond0
Ethernet Channel Bonding Driver: v3.6.0 (September 26, 2009)
Bonding Mode: adaptive load balancing
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: 0
Permanent HW addr: 74:ea:3a:6a:54:e3
Slave Interface: eth1
MII Status: up
Link Failure Count: 0
Permanent HW addr: d8:5d:4c:71:f9:94
应用拓扑：



A是双网卡绑定。
当B 发送一个arp请求到达A时，按正常情况A会回应一个arp回应报文，源mac为bond的mac，源就是bond的ip。但是这个模式下bonding驱动会截获这个arp回应，把源mac改成bond状态 下其中某一个网卡的mac:mac1，这样B收到这个arp回应时就会在arp缓存中记录下ip：1.1.1.1对应的mac为mac1。这样B的过来的流量都走MAC1.
当C 发送一个arp请求到达A时，按正常情况A会回应一个arp回应报文，源mac为bond的mac，源就是bond的ip。但是这个模式下bonding驱动会截获这个arp回应，把源mac改成bond状态 下其中某一个网卡的mac:mac2，这样C收到这个arp回应时就会在arp缓存中记录下ip：1.1.1.1对应的mac为mac2。这样C的过来的流量都走MAC2.
这样就可以做到回来让回来的流量也负载均衡。出方向均衡和MODE=5一致，不同地址会根据xor算法算出不同出口，发不同出口发送相应的arp ， mac是对应网卡的mac。