LACP Bonding in XenServer - Configuration and Troubleshooting

- CTX135690
- Created On Nov 12, 2012
- Updated On Out 03, 2013
- 27 found this helpful

- Article
- Topic : Networking

See Applicable Products

Summary

In XenServer 6.1, LACP link aggregation was added to existing bonding modes for vSwitch. This article describes dynamic link aggregation (LACP) between a XenServer host and a switch, giving a high-level overview of the 802.3ad protocol, explaining configuration and diagnosis tools.

Background

NIC Bonding

NIC bonding is a technique where two or more network cards (NICs) are configured together in order to logically function as one. It can be used for the following reasons:

- Redundancy: in case of link failure all traffic should get shifted seamlessly to the remaining NIC(s).
- Throughput aggregation: it is usually more cost-effective to bundle a few 1Gb NICs than to upgrade from 1Gb to 10Gb.

Other terms frequently used for NIC bonding are: NIC teaming, network bonding, link aggregation, link bundling or port bundling. However, it is perceived to be more correct to use link aggregation/bundling to describe configurations where set-up is required on boths sides, so both endpoints are aware of the aggregation (for example, when configuration is done on both on the server side and the switch side).

Bonding modes in XenServer 6.0

In XenServer 6.0, only two bonding modes were supported for both network stacks (Linux Bridge and vSwitch):

- *active-backup* (active-passive): only one NIC would be actively used for traffic and only in case of its failure would an inactive NIC take over,
- *balance-slb* (active-active): traffic load balancing based on the source MAC address of each Virtual Network Interface (VIF) of a guest.

Both active-backup and balance-slb do not require configuration on the switch side.

In addition to the options above, a few unsupported bonding modes existed in Linux Bridge, notably link aggregation 802.3ad.

The default bonding mode, balance-slb, has the benefit of throughput aggregation and failover, but load balancing can work well only for a sufficient number of guests(or rather VIFs), since traffic from one VIF is never split between multiple NICs. Also, its frequent rebalancing was known to cause issues in some switches (see Adjusting the bond balance interval in XenServer 6.1.0).

Bonding modes in XenServer 6.1

In XenServer 6.1, LACP support was implemented for the vSwitch network stack and LACP link aggregation was added to existing bond modes. LACP is supported only for vSwitch.

LACP bonds

Link aggregation

Link aggregation is defined as a configuration in which a group of ports is aggregated together and treated like a single interface. Its advantages are: throughput aggregation, load balancing and failover.

Static and dynamic LAG

On a switch, ports are usually grouped together by assigning them the same *LAG* (*Link Aggregation Group*) number.

There are two types of LAGs:

- *Static LAG*: ports have LACP disabled and become automatically active members of the bond. Static LAG is not widely used, as it is often considered obsolete and inferior to dynamic LAG. With static LAG on the switch, the bond mode should be `balance-slb` rather than `lacp`. Note that use of static LAG is not supported.
- *Dynamic LAG*: *Link Aggregation Control Protocol (LACP)* is used for switch-server communication, in order to negotiate dynamically which links should be active and which should be in stand-by mode.

Two names

IEEE 802.3ad, introduced in 2000, was the original standard describing link aggregation and LACP. In order to resolve a layer ordering discrepancy, it was later formally transferred to 802.1 group, becoming *IEEE 802.1AX-2008* (with no technical changes). However, the name 802.3ad remains widely used.

LACPDU frames

When using LACP protocol, both sides (the switch and the server) regularly exchange *LACPDU frames*. They contain all the information necessary to negotiate active and stand-by links, monitor the state of the links and notify the partner about a potential failure, providing a prompt failover.

Actor and Partner

Terms “Actor” and “Partner” are frequently seen when using LACP protocol. These are relative definitions: “Actor” means “this device” and “Partner” is “other device”, so the server will describe itself as the Actor with the switch being its Partner, while the switch sees itself as the Actor and refers to the server as the Partner.

Fallback to balance-slb

In the current implementation, in case of unsuccessful LACP negotiation, XenServer will automatically revert to balance-slb behavior. The server still keeps monitoring the traffic for LACP frames, so if a handshake with the switch is achieved, the bond mode will change to LACP.

Load balancing

Outgoing packets are distributed between active links. Both sides of the link — the switch and the server — are responsible for balancing their respective outgoing traffic. In XenServer, load balancing between links can be described as follows.

- A *hashing algorithm* assigns a hash number (0-255) to each packet, based on a mix of MAC, IP and TCP/UDP port as required.
- Each hash is assigned to one of the NICs on the bond, which means packets with the same hash are always sent through the same NIC.
- If a new hash is found, it is assigned to the NIC that currently has the lowest utilization.
- Rebalancing occurs at a regular interval — hashes are redistributed between the NICs to ensure all NICs are utilized to approximately the same extent.

Hashing algorithm

The hashing algorithm is chosen independently on both switch and server side, as each device balances its outgoing traffic. Rather than using the same method on both sides, hashing algorithm choice should reflect traffic patterns in each direction.

On the server side, vSwitch provides two hashing algorithms to choose from:

- `tcpudp_ports` (default), based on source and destination IP, TCP/UDP ports and source MAC;
- `src_mac`, based on source MAC address — this is the same mechanism as the one used already in balance-slb mode (traffic to/from a guest is not split).

On the switch side, the hashing algorithms depend on the switch brand and can have different names.

Use of TCP/IP parameters for hashing should provide load balancing for management traffic as well as improve it for the case of a

low number of guest, as it allows packets from the same guest to be distributed over different NICs at the same time. However, neither of the hashing algorithms presents benefits for storage traffic, as in a typical configuration large amounts of data are sent using the same source and destination IP and ports, and while the endpoints remain the same, all packets will have the same hash assigned and so will be sent through the same NIC. Consider using multipathing rather than bonding for storage traffic.

In the case of `src_mac` hashing, there is a non-negligible probability that two different MAC addresses will get the same hash number assigned — and when such “MAC clash” occurs, the two VIFs will be always using the same link. In general, `src_mac` hashing cannot provide even traffic distribution over the links if the number of VIFs is smaller than the number of NICs.

Limitations

The following limitations apply:

- LACP for Linux Bridge is not supported.
- Citrix supports bonding of up to four NICs in XenServer.
- Cisco EtherChannel is not supported.
- Cisco Port Aggregation Protocol (PAgP) is not supported.

Creating LACP bond using XenCenter

It is possible to create a LACP bond in XenCenter. The procedure is similar to creating bonds of any other type. Either go to the “Networking” tab and choose “Add Network”, and then “Bonded Network”, or alternatively, click the “Create Bond” button on the “NICs” tab. Two types of LACP bonds should be displayed: “LACP with load balancing based on IP and port of source and destination” and “LACP with load balancing based on source MAC address”. Load balancing based on IP and port of source and destination is the default hashing algorithm for LACP and it should work well for most typical configurations.

Creating LACP bond using XenServer command line

LACP bond can be created in dom0 command line as follows.

```
xe bond-create mode=lacp network-uuid=<network-uuid> pif-uuids=<pif-uuids>
```

Hashing algorithm can be specified at the creation time (default is `tcpudp_ports`):

```
xe bond-create mode=lacp properties:hashing-algorithm=<halg> network-uuid=<network-uuid>
pif-uuids=<pif-uuids>
```

where `<halg>` is `src_mac` or `tcpudp_ports`.

We can also change the hashing algorithm for an existing bond, as shown below.

```
xe bond-param-set uuid=<bond-uuid> properties:hashing_algorithm=<halg>
```

It is possible to customize the rebalancing interval by changing the bond PIF parameter `other-config:bond-rebalance-interval` and then re-plugging the PIF. The value should be expressed in millisecond. For example, following commands change the rebalancing interval to 30 seconds.

```
xe pif-param-set other-config:bond-rebalance-interval=30000 uuid=<pif-uuid>
xe pif-plug uuid=<pif-uuid>
```

The two LACP bond modes will not be displayed if you use a version older than XenCenter 6.1 or if you use Linux Bridge network stack.

Configuring LACP on a switch

Contrary to other supported bonding modes, LACP requires set-up on the switch side. The switch must support IEEE standard 802.3ad. As is the case for other bonding modes, the best practice remains to connect the NICs to different switches, in order to provide better redundancy. However, due to the setup required (with the exception of HPâs cross-switch LACP feature), all bonded NICs must be connected to the same logical switch — that is, either connected to different units of a single switch stack or connected to the same physical switch (see section “Stacked switches” in XenServer Active/Active Bonding — Switch Configuration).

There is no Hardware Compatibility List (HCL) of switches. IEEE 802.3ad is widely recognized and applied, so any switch with LACP support, as long as it observes this standard, should work with XenServer LACP bonds.

Steps for configuring LACP on a switch

1. Identify switch ports connected to the NICs to bond.
2. Using the switch web interface or command line interface, set up the same LAG (Link Aggregation Group) number for all the ports to be bonded.
3. For all the ports to be bonded set LACP to active (for example “LACP ON” or “mode auto”).
4. If necessary, bring up the LAG/port-channel interface.
5. If required, configure VLAN settings for the LAG interface — just as it would be done for a standalone port.

Example: Dell PowerConnect 6248 switch

Setting LAG 14 on port 1/g20 for Dell PowerConnect 6248 switch:

```
DELLPC-1>enable
DELLPC-1#configure
DELLPC-1(config)#interface ethernet 1/g20
DELLPC-1(config-if-1/g20)# channel-group 14 mode auto
DELLPC-1(config-if-1/g20)#exit
DELLPC-1(config)#exit
```

Bringing up the port-channel interface 14 and configuring VLAN settings (steps 4 and 5):

```
DELLPC-1>enable
DELLPC-1#configure
DELLPC-1(config)#interface port-channel 14
DELLPC-1(config-if-ch14)#switchport mode general
DELLPC-1(config-if-ch14)#switchport general pvid 1
DELLPC-1(config-if-ch14)#no switchport general acceptable-frame-type tagged-only
DELLPC-1(config-if-ch14)#switchport general allowed vlan add 1
DELLPC-1(config-if-ch14)#switchport general allowed vlan add 2-5 tagged
```

Cleaning up the above settings:

```
DELLPC-1(config-if-ch14)#no switchport mode
DELLPC-1(config-if-ch14)#no switchport general pvid
```

Example: Cisco Catalyst 3750G-A8

Configuration of LACP on ports 23 and 24 on the switch:

```
C3750-1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
C3750-1(config)#interface Port-channel3
C3750-1(config-if)#switchport trunk encapsulation dot1q
C3750-1(config-if)#switchport mode trunk
C3750-1(config-if)#exit
C3750-1(config)#interface GigabitEthernet1/0/23
C3750-1(config-if)#switchport mode trunk
C3750-1(config-if)#switchport trunk encapsulation dot1q
C3750-1(config-if)#channel-protocol lacp
C3750-1(config-if)#channel-group 3 mode active
C3750-1(config-if)#exit
C3750-1(config)#interface GigabitEthernet1/0/24
C3750-1(config-if)#switchport mode trunk
C3750-1(config-if)#switchport trunk encapsulation dot1q
C3750-1(config-if)#channel-protocol lacp
C3750-1(config-if)#channel-group 3 mode active
C3750-1(config-if)#exit
C3750-1(config)#exit
```

De-configure LACP bond on the switch side:

```
C3750-1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
C3750-1(config)#no interface port-channel3
C3750-1(config)#interface GigabitEthernet1/0/23
C3750-1(config-if)#no channel-protocol
C3750-1(config-if)#no shutdown
C3750-1(config-if)#exit
C3750-1(config)#interface GigabitEthernet1/0/24
```

```
C3750-1(config-if)#no channel-protocol
C3750-1(config-if)#no shutdown
C3750-1(config-if)#end
```

Troubleshooting

No connectivity

Lack of connectivity on the bonded network might be due to setting the LAG on the wrong ports — in case of issues, double-check that the wiring is as expected. Another reason for the lack of connectivity might be mismatched settings.

Mismatched settings

XenServer will automatically fall back to balance-slb mode if LACP is not configured on the switch. However, if LACP is set up on the switch, but not on the server, the scenario depends on the switch implementation. Many switches will drop all the traffic on the aggregated ports if LACP negotiation fails. For this reason, it is safer to create a bond on the server side first.

Bond creation taking longer

Bond creation can take more time in the case of a LACP bond (mainly due to the necessary switch-server negotiation). A short connectivity blip might be expected as well, before the desired configuration is achieved. If the set-up delay is in the order of a few seconds, this is normal and should not be of concern.

Only one link is active / not all links are active

During LACP negotiation, the switch usually has the last word in the choice of active and stand-by links. This is entirely dependent on the switch-side implementation of the protocol and the choice might be indeed different from user expectations — for example only one of three available links will be made active. In such cases, however, testing the setup might involve temporarily disabling the active link (on the server or switch side), and checking whether the correct failover occurs and other links are used.

Warning in the switch logs during LACP configuration

Some switches can issue a warning during creation of a LACP bond. If the bond eventually works, the warning should not be of concern — it is most likely issued when LACP on the server is not configured yet or if the server-side LACP bond has already reverted to balance-slb.

Diagnostics

Commands described below can help with diagnostics and troubleshooting of LACP bonds. These need to be executed in dom0, either in the XenCenter console tab or in XenServer root ssh session.

Command “xe bond-list”

XenServer command `xe bond-list` returns a list of all bonds in the pool. As for other bonding modes, it contains uuids of bond object, bond PIF and slave PIFs. Additionally, for LACP bonds the hashing algorithm is displayed in the “properties” field.

```
# xe bond-list params=all
 uuid ( RO)             : 14ebdc8c-5a21-db3c-2d72-5c66cc56075b
           master ( RO): 772829a2-a7f2-0b78-2663-344e67ecb1af
           slaves ( RO): e168b979-25e3-0640-82ab-3074f326d26e; 4d63ec5d-6341-f135-
efa8-52f22f91b8af
             mode ( RO): lacp
       properties (MRO): hashing_algorithm: tcpudp_ports
    primary-slave ( RO): 4d63ec5d-6341-f135-efa8-52f22f91b8af
         links-up ( RO): 2
```

Command “ovs-vsctl list port”

The vSwitch command `ovs-vsctl list port` returns parameters for all vSwitch ports (interfaces). The fragment of the output shown below corresponds to a LACP bond “bond0”. Note that `bond_mode` field describes the hashing algorithm rather than the XenServer bond type and a LACP bond will have the `lacp` field set to “active”.

```
# ovs-vsctl list port
```

```
[..]
_uuid               : c8dcd7e8-0972-45e0-946b-8630d8a5e7fa
bond_downdelay      : 200
bond_fake_iface     : false
bond_mode           : balance-tcp
bond_updelay        : 31000
external_ids        : {}
fake_bridge         : false
interfaces          : [49e2c5d4-077e-42fb-95cc-752782e934bc, dc16c7de-3473-4342-
b80c-625e195c02b2]
lacp                : active
mac                 : "00:24:e8:77:bd:4f"
name                : "bond0"
other_config        : {bond-detect-mode=carrier, bond-miimon-interval="100",
bond-rebalance-interval="10000"}
qos                 : []
statistics          : {}
status              : {}
tag                 : []
trunks              : []
vlan_mode           : []
[..]
```

Command “ovs-appctl bond/show”

Command `ovs-appct bond/show` returns more real-time information about the bond.

Field `bond_mode` is as above and contains the hashing algorithm currently used. The field `bond-hash-algorithm` is obsolete and is likely to be discontinued in future version of Open vSwitch. The value of `next rebalance` informs how much time is left before the hashes will be redistributed over active NICs.

Field `lacp_negotiated` is highly useful, as it indicates whether LACP negotiation with the switch was successful. This will read “false” if LACP is not set on the switch or if the negotiation did not converge for any other reason.

The second part of the output contains the list of bonded devices with all hashes currently assigned to them, as well as recent loads per hash. Observing hashes of 0kB is normal — once a hash number was assigned, it is kept in the logs even long after the relevant traffic ceased. Only rebooting the server or re-creating the bond will clean up the hash table.

```
# ovs-appctl bond/show bond0
 bond_mode: balance-tcp
 bond-hash-algorithm: balance-tcp
 bond-hash-basis: 0
 updelay: 31000 ms
 downdelay: 200 ms
 next rebalance: 8574 ms
 lacp_negotiated: true

slave eth0: enabled
     active slave
     may_enable: true
     hash 119: 16 kB load
     hash 120: 0 kB load
     hash 128: 597 kB load
     hash 132: 0 kB load
     hash 157: 5 kB load
 [..]

slave eth1: enabled
     may_enable: true
     [..]
     hash 52: 304 kB load
     hash 64: 246 kB load
     hash 74: 0 kB load
     hash 82: 0 kB load
 [..]
```

Command “ovs-appctl lacp/show”

The OVS command `ovs-appctl lacp/show` returns information related to LACP protocol — such as the negotiation status, aggregation key, LACP timeout, the identifiers of actor and key components.

```
# ovs-appctl lacp/show bond0
---- bond0 ----
    status: active negotiated
    sys_id: 00:24:e8:77:bd:4f
    sys_priority: 65534
    aggregation key: 1
    lacp_time: slow

slave: eth0: current attached
    port_id: 2
    port_priority: 65535

    actor sys_id: 00:24:e8:77:bd:4f
    actor sys_priority: 65534
    actor port_id: 2
    actor port_priority: 65535
    actor key: 1
    actor state: activity aggregation synchronized collecting distributing

    partner sys_id: 00:1c:23:6d:cd:3e
    partner sys_priority: 1
    partner port_id: 113
    partner port_priority: 1
    partner key: 646
    partner state: activity aggregation synchronized collecting distributing

slave: eth1: current attached
    port_id: 1
    port_priority: 65535

    actor sys_id: 00:24:e8:77:bd:4f
    actor sys_priority: 65534
    actor port_id: 1
    actor port_priority: 65535
    actor key: 1
    actor state: activity aggregation synchronized collecting distributing

    partner sys_id: 00:1c:23:6d:cd:3e
    partner sys_priority: 1
    partner port_id: 114
    partner port_priority: 1
    partner key: 646
    partner state: activity aggregation synchronized collecting distributing
```

Command “tcpdump”

Tools like tcpdump can be used in dom0 in order to monitor the traffic.

```
tcpdump -i eth2 -v -l
```

(press Ctrl-C to stop).

With tcpdump, we should observe LACP frames in the output.

```
09:45:21.364818 LACPv1, length 110
        Actor Information TLV (0x01), length 20
          System 00:1c:23:6d:cd:3e (oui Unknown), System Priority 1, Key 646, Port 113,
Port Priority 1
          State Flags [Activity, Aggregation, Synchronization, Collecting, Distributing]
        Partner Information TLV (0x02), length 20
          System 00:24:e8:77:bd:4f (oui Unknown), System Priority 65534, Key 1, Port 2,
Port Priority 65535
          State Flags [Activity, Aggregation, Synchronization, Collecting, Distributing]
        Collector Information TLV (0x03), length 16
          Max Delay 0
```

```
        Terminator TLV (0x00), length 0
09:45:21.894776 LACPv1, length 110
        Actor Information TLV (0x01), length 20
          System 00:24:e8:77:bd:4f (oui Unknown), System Priority 65534, Key 1, Port 2,
Port Priority 65535
          State Flags [Activity, Aggregation, Synchronization, Collecting, Distributing]
        Partner Information TLV (0x02), length 20
          System 00:1c:23:6d:cd:3e (oui Unknown), System Priority 1, Key 646, Port 113,
Port Priority 1
          State Flags [Activity, Aggregation, Synchronization, Collecting, Distributing]
        Collector Information TLV (0x03), length 16
          Max Delay 0
        Terminator TLV (0x00), length 0
```

In order to see which packets are sent on each link, multiple instances of tcpdump can be run simultaneously in separate dom0 consoles. For example, if interfaces eth0 and eth1 were bonded:

```
tcpdump -i eth0 -v
```

```
tcpdump -i eth1 -v
```

File /var/log/messages

VSwitch uses the file `/var/log/messages` to log switching traffic loads between the bonded NICs:

```
Sep  7 09:27:21 localhost ovs-vswitchd: 00130|bond|INFO|bond bond0:  shift 3kB of load
(with hash 3) from eth0 to eth1 (now carrying 1068kB  and 35kB load, respectively)
Sep  7 09:27:21 localhost ovs-vswitchd: 00131|bond|INFO|bond bond0:  shift 2kB of load
(with hash 9) from eth0 to eth1 (now carrying 1065kB  and 38kB load, respectively)
```

File /var/log/xensource.log

File `/var/log/xensource.log` contains useful network daemon entries, as well as DHCP records.

More information

If you experience any difficulties, contact Citrix Technical Support.

For a review of all supported bonding modes, refer to XenServer 6.1.0 Administrator's Guide.

Technical specification of the LACP protocol can be found in IEEE standard 802.1AX-2008.

For further information about XenServer 6.1.0 refer to XenServer 6.1.0 Release Notes.

For additional information about LACP bonding in the vSwitch network stack, refer to Open vSwitch documentation and Open vSwitch source code. XenServer 6.1.0 uses Open vSwitch 1.4.2.

Applicable Products

- XenServer 6.1.0
- XenServer 6.2.0

Share your comments or find out more about this topic

Citrix Forums