LNET Bonding

Blake Caldwell

March 2017
Network Interface Bonding

• Combining multiple network interfaces to function as a single logical interface
  – Called channel bonding
  – Dynamic link aggregation (LACP) is a specific type of bonding
  – Network teaming similar, but refers to a new driver and daemon in RHEL 7

• Implemented in Linux by the **bonding** kernel module

• Supported for Ethernet and Infiniband transport mediums
Bonding—what are the benefits?

1. High availability
   - Multiple paths (independent of Spanning Tree Protocol)

2. Increased aggregate throughput
   - Depends on hashing mode

- Can be used for Lustre servers or clients
  - Number of clients >> number of servers
Lustre Networking Layers

Lustre Client

Lustre RPCs

LNET/LND

10.0.0.1@tcp

bond0

eth0

eth1

Switch

Lustre Server

Lustre RPCs

10.0.0.2@tcp

bond0

eth0

eth1

Switch
Configuring Lustre

- Only need to specify bond interface in `lnet.conf`

  ```
  options lnet networks="tcp(bond0)"
  OR
  options lnet ip2nets="tcp0(bond0) 192.168.1.*"
  ```
Configuring Interfaces

• Modify slave interface configurations
  /etc/sysconfig/network-scripts/ifcfg-eth[0-1]
    DEVICE=eth0
    TYPE=Ethernet
    BOOTPROTO=none
    ONBOOT=yes
    MASTER=bond0
    SLAVE=yes

• Create bond interface configuration
  /etc/sysconfig/network-scripts/ifcfg-bond0
    DEVICE=bond0
    TYPE=Bond
    ONBOOT=yes
    BOOTPROTO=none
    BONDING_MASTER=yes
Ethernet Configuration

• Configuration options passed to the bonding module
  – /etc/modprobe.d/bonding.conf
  – /etc/sysconfig/network-scripts/ifcfg-bond0 BONDING_OPTS

• Example of active/active configuration:
  options bond0 mode=4 xmit_hash_policy=layer3+4

  • mode 4 specifies 802.3ad (LACP) link aggregation
  • mode 0 also supports load-balancing, but uses slave links in a round-robin fashion
  • xmit_hash_policy should be set to layer3+4 so that the hash is over source IP address, destination IP address, source port, destination port
  • default transmit hashing policy is over MAC address + packet type ID, meaning all traffic to a particular host will use the same physical link
Infiniband Configuration

- Bonding in kernel only supports active/passive
  `options bond0 mode=0`

- Alternative approach combines multi-rail and active/passive bonds for load-balancing Lustre traffic across links
  - requires Infiniband partitions configured on subnet manager
  - server-side only
Troubleshooting Tips

• Network verification is tedious
  Break one of the bonded links and test, reactivate link, break the other link, and retest

• Use the distribute option with LNET self-test to fanout connections and check for full utilization of slave links

lst add_test --distribute 1:2 --concurrency 2
Querying Bond Status

- `cat /proc/net/bonding/bond0`

  Ethernet Channel Bonding Driver: 2.6.1 (October 29, 2004)
  Bonding Mode: load balancing (round-robin)
  Currently Active Slave: eth0
  MII Status: up
  MII Polling Interval (ms): 1000
  Up Delay (ms): 0
  Down Delay (ms): 0

  Slave Interface: eth1
  MII Status: up
  Link Failure Count: 1

  Slave Interface: eth0
  MII Status: up
  Link Failure Count: 1
Tuning

• For Ethernet consider the receive buffer sizes necessary for the bandwidth of multiple links

• If the link bandwidth between Ethernet switches is the same or less than that of the host links, make sure that the switches are also hashing over active/active links

• On server-side it is possible to split the OST threads such that they are local to the NUMA node designated for a particular HCA
Conclusion

• Overview of Linux kernel bonding
• Overview of Lustre networking layers
• Configured Ethernet active/active link aggregation
• Configured Infiniband active/passive bonding
• Troubleshooting/Tuning
Resources


• Linux bonding wiki: https://wiki.linuxfoundation.org/networking/bonding

• Linux kernel documentation of bonding module: https://www.kernel.org/doc/Documentation/networking/bonding.txt


• Infiniband bonding configuration: https://community.mellanox.com/docs/DOC-2160

• Lustre interface bonding proposal: http://wiki.lustre.org/Multi-Rail_LNet/LAD15_Lustre_Interface_Bonding
This work was supported by the United States Department of Defense (DoD) and used resources of the Computational Research and Development Programs at Oak Ridge National Laboratory.