**ISCSI 磁盘在initiator映射多网卡分析**

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 版本 | 修订者 | 修订描述 |
| 2017.11.07 | v1 | 林 清 | 初稿 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 先说结论：

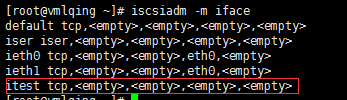
iscsi本身完全支持在单个主机上绑定多张网卡，且多张网卡的情况下网卡是同步读写的（有带宽提升）。但是从os-brick的代码分析每个compute节点的iscsi只能指定一个网卡，也就是说openstack只能使用一张网卡读写。

# ISCSI绑定多网卡

在iscsi中网卡设备是和iface绑定的，连接或discovery时可以指定iface，从而达到为某一张网卡建立到target的连接。在OpenStack中如果能够使用多个iface，结合multipath工具可以变相的拓展设备的网络带宽。

# 关于iscsiadm中的iface说明

通过命令可以查询当前主机上的所有iface信息，得到的返回信息，每一列依次表示：



|  |  |
| --- | --- |
| ***列名称*** | ***说明*** |
| iface\_name | iface的名称 |
| transport\_name | 使用的传输协议，有tcp, cxgb3i, bnx2i, iser。对于软件iscsi，指定为TCP或者iser |
| hwaddress | 硬件iscsi设备（qla4xxx）的端口MAC地址（参考man iscsiadm） |
| ipaddress | 对应的设备IP地址 |
| net\_ifacename | 软件iscsi设备可以通过指定hwaddress或者net\_ifacename来绑定物理设备，这里net\_ifacename一般为网卡名称 |
| initiatorname | iface使用的initiatorname名称。 |

安装iscsiadm之后会默认生成一个default接口，默认情形使用这个接口。default接口不和任何物理设备绑定，在工作时通过网络层为其指定一个网卡设备工作。

# 通过指定多个iface提升iscsi的带宽

测试环境由于没有多张网卡，所以两个iface绑定的是同一个网卡，但是从测试结果来看I/O确实被分成了两份，从不同的iface进行读写。

## 创建新iface接口ieth0、ieth1

     iscsiadm -m iface -I ieth0 --op=new

     iscsiadm -m iface -I ieth1 --op=new

## 为iface接口绑定网卡

这里只有一张网卡，所以全部都绑在一张上了。

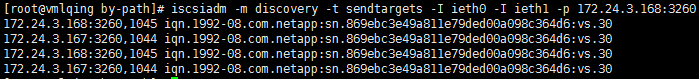
    iscsiadm -m iface -I  ieth0  --op=update -n iface.net\_ifacename -v eth0

    iscsiadm -m iface -I  ieth1  --op=update -n iface.net\_ifacename -v eth0

## 执行discovery命令发现target

    iscsiadm -m discovery -t sendtargets -I ieth0 -I ieth1 -p 172.24.3.168:3260

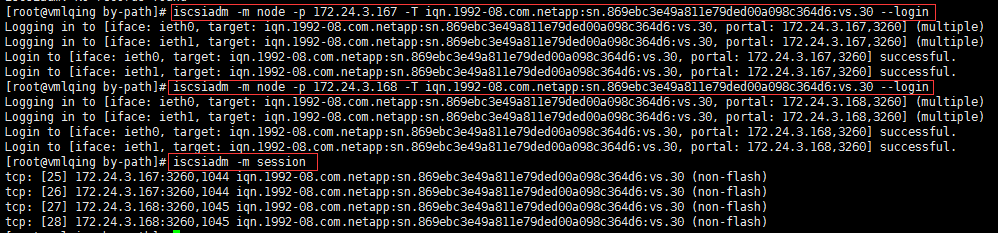
这里target侧有172.24.3.167，172.24.3.168两个IP链路，ieth0和ieth1每个iface都会发现这两个链路，所以共计四个IP链路。



## 登录所有的target连接（这里target=ip+iqn）

iscsiadm -m node –p 172.24.3.167 -T iqn.1992-08.com.netapp:sn.869ebc3e49a811e79ded00a098c364d6:vs.30 --login

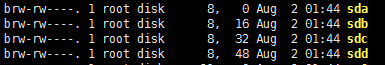
iscsiadm -m node -p 172.24.3.168 -T iqn.1992-08.com.netapp:sn.869ebc3e49a811e79ded00a098c364d6:vs.30 --login



可以看出ieth0和ieth1每个iface都会登录，最终建立四个session。

## 映射netapp的LUN到这个主机

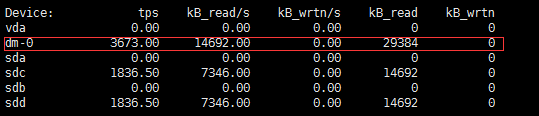
可以看到最终四个Session映射出四个磁盘设备，四个设备对应的多路径融合设备为/dev/dm-0



其中，iface和磁盘设备的关系如下：

|  |  |  |
| --- | --- | --- |
|  | ieth0 | ieth1 |
| 172.24.3.167 | sdd | sdc |
| 172.24.3.168 | sda | sdb |

向四个磁盘设备聚合的虚拟设备/dev/dm-0写数据，发现multipath可以将读写分摊到不同的iface中提高读写效率，这里实验时由于没有两个网卡所以两个iface配置的是同一张网卡。但是对于同一个iface中的不同链路，始终是主备的读写方式。



# 在nova中配置多个iface的可行性

nova\_compute在读写iscsi磁盘时默认使用default接口。

分析iscsi\_iface的配置项发现，该配置项可以修改Openstack使用的iface。但是根据配置项说明，替换的iface必须是be2iscsi, bnx2i, cxgb3i, cxgb4i, qla4xxx, ocs类型之一，这些类型都是硬件iscsi设备的类型，软件iscsi协议的transport\_name值为tcp。所以可以推测iscsi\_iface配置项是用来配置硬件iscsi设备时使用的，不能实现配置多个iface。

cfg.StrOpt('iscsi\_iface', deprecated\_name='iscsi\_transport',

help="""

The iSCSI transport iface to use to connect to target in case offload support is desired.

Default format is of the form <transport\_name>.<hwaddress> where

<transport\_name> is one of (be2iscsi, bnx2i, cxgb3i, cxgb4i, qla4xxx, ocs) and

<hwaddress> is the MAC address of the interface and can be generated via the

iscsiadm -m iface command. Do not confuse the iscsi\_iface parameter to be

provided here with the actual transport name.

""")

分析os-brick代码，os-brick中iscsi的connector初始化时，会对指定的iface名称进行检查，检查代码如下：

supported\_transports = ['be2iscsi', 'bnx2i', 'cxgb3i', 'default',

'cxgb4i', 'qla4xxx', 'ocs', 'iser']

def \_validate\_iface\_transport(self, transport\_iface):

"""Check that given iscsi\_iface uses only supported transports

Accepted transport names for provided iface param are

be2iscsi, bnx2i, cxgb3i, cxgb4i, default, qla4xxx, ocs or iser.

Note the difference between transport and iface;

unlike default(iscsi\_tcp)/iser, this is not one and the same for

offloaded transports, where the default format is

transport\_name.hwaddress

:param transport\_iface: The iscsi transport type.

:type transport\_iface: str

:returns: str

"""

# Note that default(iscsi\_tcp) and iser do not require a separate

# iface file, just the transport is enough and do not need to be

# validated. This is not the case for the other entries in

# supported\_transports array.

if transport\_iface in ['default', 'iser']:

return transport\_iface

# Will return (6) if iscsi\_iface file was not found, or (2) if iscsid

# could not be contacted

out = self.\_run\_iscsiadm\_bare(['-m','iface','-I',transport\_iface],check\_exit\_code=[0, 2, 6])[0] or ""

LOG.debug("iscsiadm %(iface)s configuration: stdout=%(out)s.",

{'iface': transport\_iface, 'out': out})

for data in [line.split() for line in out.splitlines()]:

if data[0] == 'iface.transport\_name':

if data[2] in self.supported\_transports:

return transport\_iface

LOG.warning("No useable transport found for iscsi iface %s. "

"Falling back to default transport.",

transport\_iface)

return 'default'