# 背景

由于多路径软件是需要和存储在一起配合使用的，不同的厂商基于不同的操作系统，都提供了不同的版本。并且有的厂商，软件和硬件也不是一起卖的，如果要使用多路径软件的话，可能还需要向厂商购买license才行。比如EMC公司基于Linux下的多路径软件，就需要单独的购买license。其中，EMC提供的就是PowerPath，HDS提供的就是HDLM，Veritas提供的就是VxDMP。当然，使用系统自带的免费多路径软件包，同时也是一个比较通用的包，可以支持大多数存储厂商的设备，即使是一些不是出名的厂商，通过对配置文件进行稍作修改，也是可以支持并运行的很好。

比较重要的一点还是听从原厂工程师的建议根据实际的业务和存储策略使用合适的多路径软件。

# 概述

Multipath即多路径，是个通用概念。这里要介绍的是开源的存储多路径技术，也就是DM multipath。

普通的电脑主机都是一个硬盘挂接到一个总线上，这里是一对一的关系。而到了有光纤组成的SAN环境，由于主机和存储通过了光纤交换机连接，这样的话，就构成了多对多的关系。也就是说，主机到存储可以有多条路径可以选择。主机到存储之间的IO由多条路径可以选择。既然，每个主机到所对应的存储可以经过几条不同的路径，如果是同时使用的话，I/O流量如何分配？其中一条路径坏掉了，如何处理？还有在操作系统的角度来看，每条路径，操作系统会认为是一个实际存在的物理盘，但实际上只是通向同一个物理盘的不同路径而已，这样是在使用的时候，就给用户带来了困惑。

多路径软件就是为了解决上面的问题应运而生的，多路径的主要功能就是和存储设备一起配合实现如下功能：

1. 故障的切换和恢复
2. IO流量的负载均衡
3. 磁盘的虚拟化

# Device Mapper

# 命名

有时看到一串16进制数字(WWID)，有时是以mpath为前缀的名字(user-friendly name)，有时是任意字母串(alia name)。

multipath默认用的是WWID，为什么不用好记的名字呢？好记的名字不能工作的一个情景：根文件系统不能在multipath设备上面。好记的名字和 WWID之间的映射是保存在/etc/multipath/bindings文件里的。要访问这个文件，根文件系统必须已经挂载上了，而multipath服务在initrd里就要开始工作，那个时候还没有根系统。因此，默认设置为wwid是为了安全。

# 设备地址

这种以冒号分割4个数字的设备地址如2:0:0:1怎么解释？

2:0:0:1设备地址，数字分别对应：Host:Bus:Target:Lun。比如我们让iscsi target走了两个IP地址，那么对于同一个设备只有 host字段不同。比如：2:0:0:1和3:0:01。

# 操作

完整操作：

#查看主机或者存储交换机上的WWN号，在存储上将LUN映射给需要的主机

cat /sys/class/fc\_host/host\*/port\_name

0x2002d0431efb7f5d

0x2001d0431efb7f5d

#在系统内执行扫盘命令，没有命令先安装sg3\_utils

yum install sg3\_utils

rescan-scsi-bus.sh

#查看是否映射到对应的

fdisk -l

lsblk

#查看是否安装了multipath

yum install device-mapper-multipath

rpm -qa | grep device-mapper

device-mapper-libs-1.02.107-5.el7\_2.2.x86\_64

device-mapper-persistent-data-0.5.5-1.el7.x86\_64

device-mapper-multipath-0.4.9-85.el7\_2.4.x86\_64

device-mapper-1.02.107-5.el7\_2.2.x86\_64

device-mapper-event-libs-1.02.107-5.el7\_2.2.x86\_64

device-mapper-event-1.02.107-5.el7\_2.2.x86\_64

device-mapper-multipath-libs-0.4.9-85.el7\_2.4.x86\_64

#拷贝默认的multipath.conf到/etc目录下，也可以使用mpathconf命令创建默认模板

cp /usr/share/doc/device-mapper-multipath-0.4.9/multipath.conf /etc/multipath.conf

mpathconf --enable --with\_multipathd y

#查看本地存储wwid

/lib/udev/scsi\_id --whitelisted --device=/dev/sda

36141877030bcca001d9c4d52106b9d90

#查看存储参数

cat /sys/block/sdb/device/vendor

3PARdata

cat /sys/block/sdb/device/model

VV

#添加本地盘到blacklist内,不同的存储和系统参考官方的最佳实践

vim /etc/multipath.conf

defaults {

polling\_interval 10

user\_friendly\_names no

find\_multipaths yes

}

blacklist {

wwid "<wwid\_of\_the\_local\_disk>"

}

devices {

device {

vendor "3PARdata"

product "VV"

path\_grouping\_policy group\_by\_prio

path\_selector "round-robin 0"

path\_checker tur

features "0"

hardware\_handler "1 alua"

prio alua

failback immediate

rr\_weight uniform

no\_path\_retry 18

rr\_min\_io\_rq 1

detect\_prio yes

# fast\_io\_fail\_tmo 10

# dev\_loss\_tmo 14

}

}

#固定设备的别名

cat /etc/multipath/bindings

# Multipath bindings, Version : 1.0

# NOTE: this file is automatically maintained by the multipath program.

# You should not need to edit this file in normal circumstances.

#

# Format:

# alias wwid

#

mpatha 360002ac000000000000000260001a945

mpathb 360002ac000000000000000270001a945

mpathc 360002ac000000000000000280001a945

mpathd 360002ac000000000000000290001a945

mpathe 360002ac0000000000000002a0001a945

#编辑multipath.conf，增加以下字段

vim /etc/multipath.conf

multipaths {

multipath {

wwid 360002ac000000000000000260001a945

alias 3PAR8400\_DEV\_LUN26

}

multipath {

wwid 360002ac000000000000000270001a945

alias 3PAR8400\_DEV\_LUN27

}

multipath {

wwid 360002ac000000000000000280001a945

alias 3PAR8400\_DEV\_LUN28

}

multipath {

wwid 360002ac000000000000000290001a945

alias 3PAR8400\_DEV\_LUN29

}

multipath {

wwid 360002ac0000000000000002a0001a945

alias 3PAR8400\_DEV\_LUN30

}

}

#清空已有的multipath记录

multipath -F

#打印诊断信息

multipath -v3

#启用多路径守护程序以在引导时启动

mpathconf --enable

#启动多路径服务

service multipathd start

#如果在启动 multipath 守护程序后更改多路径配置文件，请运行以下命令以使更改生效。

service multipathd reload

#重启系统测试

init 6

#查看多路径当前状态

multipath -ll

#Enable extended logging on the qla2xxx driver

chmod u+x /sys/module/qla2xxx/parameters/ql2xextended\_error\_logging

echo "1" > /sys/module/qla2xxx/parameters/ql2xextended\_error\_logging

cat /sys/module/qla2xxx/parameters/ql2xextended\_error\_logging

#重启lvm服务是识别共享存储vg信息

systemctl restart lvm2-lvmetad.service

pvs

#屏蔽WARNING: duplicate提示信息

vim /etc/lvm/lvm.conf

# Configuration option devices/filter.

# Limit the block devices that are used by LVM commands.

# This is a list of regular expressions used to accept or reject block

# device path names. Each regex is delimited by a vertical bar '|'

# (or any character) and is preceded by 'a' to accept the path, or

# by 'r' to reject the path. The first regex in the list to match the

# path is used, producing the 'a' or 'r' result for the device.

# When multiple path names exist for a block device, if any path name

# matches an 'a' pattern before an 'r' pattern, then the device is

# accepted. If all the path names match an 'r' pattern first, then the

# device is rejected. Unmatching path names do not affect the accept

# or reject decision. If no path names for a device match a pattern,

# then the device is accepted. Be careful mixing 'a' and 'r' patterns,

# as the combination might produce unexpected results (test changes.)

# Run vgscan after changing the filter to regenerate the cache.

# See the use\_lvmetad comment for a special case regarding filters.

#

# Example

# Accept every block device:

# filter = [ "a|.\*/|" ]

# Reject the cdrom drive:

# filter = [ "r|/dev/cdrom|" ]

# Work with just loopback devices, e.g. for testing:

# filter = [ "a|loop|", "r|.\*|" ]

# Accept all loop devices and ide drives except hdc:

# filter = [ "a|loop|", "r|/dev/hdc|", "a|/dev/ide|", "r|.\*|" ]

# Use anchors to be very specific:

# filter = [ "a|^/dev/hda8$|", "r|.\*/|" ]

#

# This configuration option has an automatic default value.

# filter = [ "a|.\*/|" ]

filter = ["a/sda/","a/mapper/3PAR8400\_DEV\_LUN26p1/","r/.\*/"]

#filter = ["a/sda/","r/.\*/"]

# Configuration option devices/global\_filter.

# Limit the block devices that are used by LVM system components.

# Because devices/filter may be overridden from the command line, it is

# not suitable for system-wide device filtering, e.g. udev and lvmetad.

# Use global\_filter to hide devices from these LVM system components.

# The syntax is the same as devices/filter. Devices rejected by

# global\_filter are not opened by LVM.

# This configuration option has an automatic default value.

# global\_filter = [ "a|.\*/|" ]

global\_filter = ["a/sda/","a/mapper/3PAR8400\_DEV\_LUN26p1/","r/.\*/"]

#一个简单的multipath.conf配置文件

[root@controller01 ~]# cat /etc/multipath.conf

# This is a basic configuration file with some examples, for device mapper

# multipath.

#

# For a complete list of the default configuration values, run either

# multipath -t

# or

# multipathd show config

#

# For a list of configuration options with descriptions, see the multipath.conf

# man page

## By default, devices with vendor = "IBM" and product = "S/390.\*" are

## blacklisted. To enable mulitpathing on these devies, uncomment the

## following lines.

#blacklist\_exceptions {

# device {

# vendor "IBM"

# product "S/390.\*"

# }

#}

## Use user friendly names, instead of using WWIDs as names.

defaults {

user\_friendly\_names yes

find\_multipaths yes

}

##

## Here is an example of how to configure some standard options.

##

#

#defaults {

# polling\_interval 10

# path\_selector "round-robin 0"

# path\_grouping\_policy multibus

# uid\_attribute ID\_SERIAL

# prio alua

# path\_checker readsector0

# rr\_min\_io 100

# max\_fds 8192

# rr\_weight priorities

# failback immediate

# no\_path\_retry fail

# user\_friendly\_names yes

#}

##

## The wwid line in the following blacklist section is shown as an example

## of how to blacklist devices by wwid. The 2 devnode lines are the

## compiled in default blacklist. If you want to blacklist entire types

## of devices, such as all scsi devices, you should use a devnode line.

## However, if you want to blacklist specific devices, you should use

## a wwid line. Since there is no guarantee that a specific device will

## not change names on reboot (from /dev/sda to /dev/sdb for example)

## devnode lines are not recommended for blacklisting specific devices.

##

#blacklist {

# wwid 26353900f02796769

# devnode "^(ram|raw|loop|fd|md|dm-|sr|scd|st)[0-9]\*"

# devnode "^hd[a-z]"

#}

#multipaths {

# multipath {

# wwid 3600508b4000156d700012000000b0000

# alias yellow

# path\_grouping\_policy multibus

# path\_selector "round-robin 0"

# failback manual

# rr\_weight priorities

# no\_path\_retry 5

# }

# multipath {

# wwid 1DEC\_\_\_\_\_321816758474

# alias red

# }

#}

#devices {

# device {

# vendor "COMPAQ "

# product "HSV110 (C)COMPAQ"

# path\_grouping\_policy multibus

# path\_checker readsector0

# path\_selector "round-robin 0"

# hardware\_handler "0"

# failback 15

# rr\_weight priorities

# no\_path\_retry queue

# }

# device {

# vendor "COMPAQ "

# product "MSA1000 "

# path\_grouping\_policy multibus

# }

#}

multipaths {

multipath {

wwid 360002ac000000000000000260001a945

alias 3PAR8400\_DEV\_LUN26

}

multipath {

wwid 360002ac000000000000000270001a945

alias 3PAR8400\_DEV\_LUN27

}

multipath {

wwid 360002ac000000000000000280001a945

alias 3PAR8400\_DEV\_LUN28

}

multipath {

wwid 360002ac000000000000000290001a945

alias 3PAR8400\_DEV\_LUN29

}

multipath {

wwid 360002ac0000000000000002a0001a945

alias 3PAR8400\_DEV\_LUN30

}

multipath {

wwid 360002ac0000000000000002b0001a945

alias 3PAR8400\_DEV\_LUN31

}

multipath {

wwid 360002ac0000000000000002c0001a945

alias 3PAR8400\_DEV\_LUN32

}

multipath {

wwid 360002ac0000000000000002d0001a945

alias 3PAR8400\_DEV\_LUN33

}

multipath {

wwid 360002ac0000000000000002e0001a945

alias 3PAR8400\_DEV\_LUN34

}

multipath {

wwid 360002ac0000000000000002f0001a945

alias 3PAR8400\_DEV\_LUN35

}

multipath {

wwid 360002ac000000000000000320001a945

alias 3PAR8400\_DEV\_LUN36

}

multipath {

wwid 360002ac000000000000000330001a945

alias 3PAR8400\_DEV\_LUN37

}

multipath {

wwid 360002ac000000000000000340001a945

alias 3PAR8400\_DEV\_LUN38

}

multipath {

wwid 360002ac000000000000000350001a945

alias 3PAR8400\_DEV\_LUN39

}

multipath {

wwid 360002ac000000000000000360001a945

alias 3PAR8400\_DEV\_LUN40

}

multipath {

wwid 360002ac000000000000000370001a945

alias 3PAR8400\_DEV\_LUN41

}

multipath {

wwid 360002ac000000000000000380001a945

alias 3PAR8400\_DEV\_LUN42

}

multipath {

wwid 360002ac000000000000000390001a945

alias 3PAR8400\_DEV\_LUN43

}

multipath {

wwid 360002ac0000000000000003a0001a945

alias 3PAR8400\_DEV\_LUN44

}

multipath {

wwid 360002ac0000000000000003b0001a945

alias 3PAR8400\_DEV\_LUN45

}

multipath {

wwid 360002ac0000000000000003d0001a945

alias 3PAR8400\_DEV\_LUN46

}

multipath {

wwid 360002ac0000000000000003e0001a945

alias 3PAR8400\_DEV\_LUN47

}

multipath {

wwid 360002ac0000000000000003f0001a945

alias 3PAR8400\_DEV\_LUN48

}

multipath {

wwid 360002ac000000000000000400001a945

alias 3PAR8400\_DEV\_LUN49

}

multipath {

wwid 360002ac000000000000000410001a945

alias 3PAR8400\_DEV\_LUN50

}

}

## 基本指令

multipath

modprobe dm-multipath

service multipathd reload （start,stop,restart）

multipath -v2

multipath -v3 格式化路径 -v3 更详细

multipath -F

multipath -ll

multipathd -k (help)

注意：由于配置multipath 后，在新添加的本地磁盘或存储阵列的lun时，需要考虑blacklist的和LVM配置文件对磁盘的过滤问题，不正确的参数可能导致主机无非正常识别和使用新添加的磁盘,需要视情况修改。

两个选项：

file:/etc/lvm/lvm.conf

filter = [ "a/.\*/" ]

file:/etc/multipath.conf

blacklist {

devnode "^sda"

}

## 操作步骤

1、安装rpm包；

2、启动multipath：

将多路径软件添加至内核模块中

modprobe dm-multipath

modprobe dm-round-robin

检查内核模块加载状况：

lsmod | grep multipath

将多路径软件multipath设置为开机自启动：

chkconfig –level 2345 multipath on

或

chkconfig multipathd on

chkconfig –list | grep multipathd

启动multipath服务：

service multipathd restart

3、多路径配置：

停止服务：

service multipathd stop

获取所有存储LUN的WWID：

创建配置文件/etc/multipath.conf：

/sbin/,ultipathconf –enable –find\_multipaths y –with\_module y –with\_chkconfig y

多路径配置文件中添加内容：

blacklist {

devnode "本地磁盘" ##本地磁盘不需要多路径，加入黑名单

}

multipaths{

multipath{

wwid

alias #设置数据盘、日志盘等

}

}

# path group

什么是path group?

path grouping policy and IO scheduling policy?

起初，我对这个概念有混淆：认为一个真实设备对应的所有路径为一个path group，即认为下面是一个path group:

multipath-demo:~ # multipath -l14945540000000000ccb70d0ceeee4280f8450284d6298b59 dm-0 IET,VIRTUAL-DISK

size=10G features='1 retain\_attached\_hw\_handler' hwhandler='0' wp=rw

|-+- policy='service-time 0' prio=0 status=active

| `- 2:0:0:0 sda 8:0  active undef unknown`-+- policy='service-time 0' prio=0 status=enabled

 `- 3:0:0:0 sdc 8:32 active undef unknown

其实，dm-0设备有两个path group，每个PG都只有一个路径（真实环境有多条），状态active的是正在工作的路径，状态enabled处于备用状态，并不下发IO。

path grouping policy 默认是failover, 如martin所说，各设备厂商默认策略不同，主流的在用group\_by\_prio，作用就是把路径分组。IO scheduling policy默认是service time, 负责如何在一个PG的路径中分配IO。

# 应用

在没有企业存储的情况下，怎么玩multipath？

使用虚拟机和iscsi。装一虚拟机，添加块设备，添加两个网卡，再用这个块设备建一个iscsi target。然后在一个想玩multipath的机器上面，用iscsi client去连接iscsi target。至此，用lsblk会查看到原来的块设备有两个设备节点。