## 案例九：监控网卡流量

在我的运维生涯中，曾遇到过一个很奇怪的问题，一台服务器运行一段时间后网卡流量变为0，也就是说网卡无法传输数据了。但是重启网络服务后恢复正常。造成该问题的原因可能是系统内核的问题，也可能是网卡硬件问题。

当时我的思路是，先尝试重新安装操作系统，看问题是否可以解决，如果问题依然存在然后尝试更换网卡。由于服务器上的业务不能中断，所以重装操作系统或者更换网卡都无法在近期内实现。临时解决办法是写一个shell脚本监控网卡流量，当流量为0时，重启网络服务。

本案例具体需求如下：

1）一分钟监控一次网卡流量

2）当网卡流量为0时，重启网卡

3）假设网卡名字为eth0

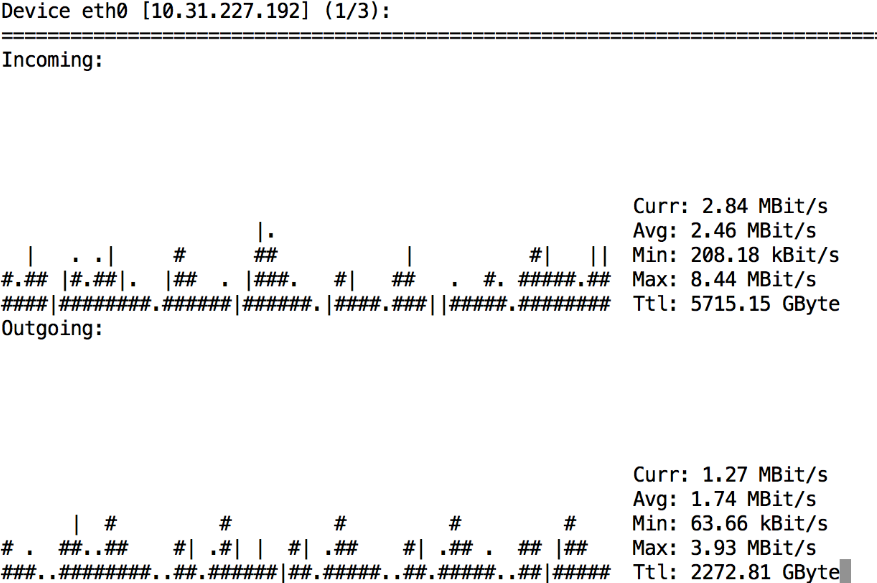
知识点一：查看网卡流量

在案例八中，我们使用sar -n DEV查看网卡流量，其实还有一种更直观的方法，那就是使用nload查看网卡流量。要使用该工具，需要先安装nload包，在CentOS系统中，需要这样安装：

# yum install -y epel-release

# yum install -y nload

安装好nload包之后，输入nload命令就可以动态查看网卡流量了，如图所示：



如果想要查看另外一块网卡的流量，需要按一下向右方向键即可。使用nload -m可以同时显示全部网卡的实时流量。但是，nload这个工具在shell脚本中无法使用，因为它是动态的。

知识点二：重启网络服务

CentOS系统，查看网卡和IP的命令是ip addr，如下：

# ip addr

1: lo: &lt;LOOPBACK,UP,LOWER\_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1

link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

inet 127.0.0.1/8 scope host lo

valid\_lft forever preferred\_lft forever

2: eth0: &lt;BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP qlen 1000

link/ether 00:16:3e:05:c6:18 brd ff:ff:ff:ff:ff:ff

inet 10.31.227.192/22 brd 10.31.227.255 scope global eth0

valid\_lft forever preferred\_lft forever

3: eth1: &lt;BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP qlen 1000

link/ether 00:16:3e:05:26:b4 brd ff:ff:ff:ff:ff:ff

inet 105.28.145.126/22 brd 105.28.147.255 scope global eth1

valid\_lft forever preferred\_lft forever

网卡配置文件所在路径：/etc/sysconfig/network-scripts/，在这个目录下有几个带有网卡名字的配置文件，例如ifcfg-eth0，其中eth0网卡配置文件内容如下：

＃ cat /etc/sysconfig/network-scritps/ifcfg-eth0

DEVICE=eth0

ONBOOT=yes

BOOTPROTO=static

IPADDR=10.31.227.192

NETMASK=255.255.252.0

要想更改IP地址，修改对应的配置文件即可，修改完之后需要重启网络服务：

# systemctl restart network //CentOS7系统

如果是CentOS6系统，命令是这样的：

# service network restart

也可以单独重启指定网卡，例如针对eth0，命令如下：

# ifdown eth0

# ifup eth0

知识点三：抓包

在平时的日常运维工作中，如果流量异常（通常是流量变大）需要我们分析到底是什么样的流量，而抓包工具用来完成该需求。在CentOS系统里有一个常用的抓包工具tcpdump，若系统没有这个命令，需安装一下tcpdump包，命令为：

# yum install -y tcpdump

我最常用的tcpdump用法是：

# tcpdump -nn -i eth0 -c 100

说明：-i选项指定网卡名字，-c选项指定抓包的数量，-nn选项的作用是以数字的形式显示IP和Port。该用法在屏幕上显示数据包的流向，即显示来源IP、端口和目标IP、端口。要想抓到真正的数据包，需加上-w选项。

# tcpdump -nn -i eth0 -c 100 -w /tmp/1.cap

这样会抓100个数据包并存入/tmp/1.cap文件里，这个文件是二进制的，可以用windows版的wireshark工具（图形化）查看也可以用tcpdump查看：

# tcpdump -r /tmp/1.cap

另外，tcpdump也可以指定IP、Port以及协议：

# tcpdump -nn tcp and host 1.1.1.1 and port 80

知识点四：if判断多个条件

在shell脚本中，if判断条件经常会有多个，要么同时满足要么只满足一个。用法如下：

# if [ option1 -a option2 ] //同时满足option1和option2

# if [ option1 ] && [ option2 ] //同时满足option1和option2

# if [ option1 -o option2 ] //满足option1或option2

# if [ option1 ] || [ option2 ] //满足option1或option2

本案例参考脚本

#!/bin/bash

#监控网卡流量，当流量为0，重启网卡

#作者：

#日期：2018-10-07

#版本：v0.2

#设定语言为英文

LANG=en

#判定系统是否已经安装sysstat包，该包里有sar命令

if ! rpm -q sysstat &>/dev/null

then

yum install -y sysstat

fi

#将10秒的网卡流量写入到一个临时文件里

sar -n DEV 1 10 |grep 'eth0' > /tmp/eth0\_sar.log

#入口网卡流量

net\_in=`grep '^Average:' /tmp/eth0\_sar.log|awk '{print $5}'`

#出口网卡流量

net\_out=`grep '^Average:' /tmp/eth0\_sar.log|awk '{print $6}'`

#当入口和出口流量同时为0时，说明网卡异常

if [ $net\_in == "0.00" -a $net\_out == "0.00" ]

then

echo "`date` eth0网卡出现异常，重启网卡。">> /tmp/net.log

ifdown eth0 && ifup eth0

fi

一分钟执行一次，写一个计划任务即可。

需求扩展

网卡出问题，流量跌为0的情况非常罕见，但流量突然大幅增涨的情况倒是很普遍。现在的需求是，当网卡（eth0）流量增幅超过1倍时需要抓1000个数据包存入一个以日期、时间为名字的文件中，并且需要发邮件给admin@admin.com(不考虑告警收敛)。

参考脚本，如下：

#!/bin/bash

#监控网卡流量增幅超过一倍告警

#作者：

#日期：2018-10-07

#版本：v0.1

mail\_user=admin@admin.com

dir=/tmp/netlog

[ -d $dir ] || mkdir $dir

s\_m=`lsattr -d $dir|awk '{print $1}' |sed 's/[^a]//g'`

if [ $s\_m != "a" ]

then

chattr $dir

fi

if ! rpm -q sysstat &>/dev/null

then

yum install -y sysstat

fi

sar -n DEV 1 10 |grep 'eth0' > /tmp/eth0\_sar.log

net\_in=`grep '^Average:' /tmp/eth0\_sar.log|awk '{print $5}'`

net\_out=`grep '^Average:' /tmp/eth0\_sar.log|awk '{print $6}'`

if [ ! -f $dir/net.log ]

then

echo "net\_in $net\_in" >> $dir/net.log

echo "net\_out $net\_out" >> $dir/net.log

exit 0

fi

net\_in\_last=`tail -2 $dir/net.log|grep 'net\_in'`

net\_out\_last=`tail -2 $dir/net.log|grep 'net\_out'`

net\_in\_diff=`$[$net\_in-$net\_in\_last]`

net\_out\_diff=`$[$net\_out-$net\_out\_last]`

if [ $net\_in\_diff -gt $net\_in\_last ]

then

python mail.py $mail\_user "网卡入口流量增幅异常" "增幅$net\_in\_diff"

#这里的mail.py参考案例二的知识点五

fi

if [ $net\_out\_diff -gt $net\_out\_last ]

then

python mail.py $mail\_user "网卡出口流量增幅异常" "增幅$net\_out\_diff"

fi

echo "net\_in $net\_in" >> $dir/net.log

echo "net\_out $net\_out" >> $dir/net.log