

环境配置

1.熟悉centos linux主机

2.配置gcc环境

3.分别配置nginx、lvs环境

3.1 nginx

先安装3个依赖

依赖1: pcre:

```
1 cd /usr/local/src
2
3 wget http://downloads.sourceforge.net/project/pcre/pcre/8.35/pcre-8.35.tar.gz
4
5 tar -zxvf pcre-8.35.tar.gz -C ./
6
7 cd pcre-8.35
8
9 ./configure
10
11 make
12
13 make install
```

依赖2: zlib:

```
1 cd /usr/local/src
2
3 wget http://zlib.net/zlib-1.2.11.tar.gz
4
5 tar -zxvf zlib-1.2.11.tar.gz -C ./
6
7 cd zlib-1.2.11
8
9 ./configure
10
11 make
12
13 make install
```

依赖3: openssl:

```
1 cd /usr/local/src
2
3 wget https://www.openssl.org/source/openssl-1.0.2s.tar.gz
4
5 tar -zxvf openssl-1.0.2s.tar.gz -C ./
6
7 cd openssl-1.0.2s
8
9 ./config
10
11 make
12
13 make install
```

nginx安装

```
1 cd /usr/local/src
2
3 wget http://nginx.org/download/nginx-1.12.1.tar.gz*
4
5 tar -zxvf nginx-1.12.1.tar.gz -C ./
6
7 cd nginx-1.12.1
8
9 ./configure --sbin-path=/usr/local/nginx/nginx --conf-path=/usr/local/nginx/nginx.conf
  --pid-path=/usr/local/nginx/nginx.pid --with-http_ssl_module --with-
  pcre=/usr/local/src/pcre-8.35 --with-zlib=/usr/local/src/zlib-1.2.11 --with-
  openssl=/usr/local/src/openssl-1.0.2s
10
11 make
12
13 make install
```

注:

完整脚本

installNginx.sh

```
1 cd / && \
2 cd /usr/local/src && \
3 wget http://downloads.sourceforge.net/project/pcre/pcre/8.35/pcre-8.35.tar.gz && \
4 tar -zxvf pcre-8.35.tar.gz -C ./ && \
5 cd pcre-8.35 && \
6 ./configure && make && make install && echo "pcre安装完毕" && \
7 cd .. && \
8 wget http://zlib.net/zlib-1.2.11.tar.gz && \
9 tar -zxvf zlib-1.2.11.tar.gz -C ./ && \
10 cd zlib-1.2.11 && \
11 ./configure && make && make install && echo "zlib安装完毕" && \
12 cd .. && \
13 wget https://www.openssl.org/source/openssl-1.0.2s.tar.gz && \
14 tar -zxvf openssl-1.0.2s.tar.gz -C ./ && \
15 cd openssl-1.0.2s && \
16 ./config && make && make install && echo "openssl安装完毕" && \
17 cd .. && \
18 wget http://nginx.org/download/nginx-1.12.1.tar.gz && \
19 tar -zxvf nginx-1.12.1.tar.gz -C ./ && \
20 cd nginx-1.12.1 && \
21 ./configure \
22 --sbin-path=/usr/local/nginx/nginx \
23 --conf-path=/usr/local/nginx/nginx.conf \
24 --pid-path=/usr/local/nginx/nginx.pid \
25 --with-http_ssl_module \
26 --with-pcre=/usr/local/src/pcre-8.35 \
27 --with-zlib=/usr/local/src/zlib-1.2.11 \
28 --with-openssl=/usr/local/src/openssl-1.0.2s && \
```

```
29 | make && make install && echo "nginx安装完毕"
```

安装完毕后的启动路径

```
1 | /usr/local/src/nginx-1.12.1/objs/nginx
```

先查看80端口是否被使用

```
1 | netstat -ano|grep 80 # 检查80端口是否被使用
```

到对应路径使用

```
1 | ./nginx # 启动
2
3 | ./nginx -s stop # 关闭
4
5 | ./nginx -s reload # 重启
```

4.lvs-fullnat版（小坑怡情）

1.安装包准备

```
1 | kernel-2.6.32-220.23.1.el6.src.rpm
2
3 | asciidoc-8.4.5-4.1.el6.noarch.rpm
4
5 | newt-devel-0.52.11-3.el6.x86_64.rpm
6
7 | slang-devel-2.2.1-1.el6.x86_64.rpm
8
9 | Lvs-fullnat-synproxy.tar.gz
```

2.安装一下rpm包

```
1 # 建内核目录工具
2 yum install rpmdevtools yum-utils
3
4 # 自动生成内核目录结构
5 rpmdev-setuptree
6
7 # 解压安装
8 yum-builddep kernel-<version>.src.rpm
9
10 rpm -Uvh kernel-<version>.src.rpm
```

3.准备内核源码树

```
1 cd ~/rpmbuild/SPECS
2
3 rpmbuild -bp --target=$(uname -m) kernel.spec
```

```
1 # 1号陨石坑----缺少一部分依赖
2 error: Failed build dependencies:
3     gcc >= 3.4.2 is needed by kernel-2.6.32-220.23.1.el6.x86_64
4     redhat-rpm-config is needed by kernel-2.6.32-220.23.1.el6.x86_64
5     patchutils is needed by kernel-2.6.32-220.23.1.el6.x86_64
6     xmlto is needed by kernel-2.6.32-220.23.1.el6.x86_64
7     asciidoc is needed by kernel-2.6.32-220.23.1.el6.x86_64
8     elfutils-libelf-devel is needed by kernel-2.6.32-220.23.1.el6.x86_64
9     zlib-devel is needed by kernel-2.6.32-220.23.1.el6.x86_64
10    binutils-devel is needed by kernel-2.6.32-220.23.1.el6.x86_64
11    newt-devel is needed by kernel-2.6.32-220.23.1.el6.x86_64
12    python-devel is needed by kernel-2.6.32-220.23.1.el6.x86_64
13    perl(ExtUtils::Embed) is needed by kernel-2.6.32-220.23.1.el6.x86_64
14    hmacalc is needed by kernel-2.6.32-220.23.1.el6.x86_64
```

解决方法

```
1 # 安装完毕后重新运行 rpmbuild -bp --target=$(uname -m) kernel.spec
2 yum install -y gcc asciidoc newt-devel
3
4 yum install redhat-rpm-config patchutilsxmlto elfutils-libelf-devel binutils-devel
5 python-devel hmacalcp Perl-ExtUtils-Embed -y
6
6 yum install perl-ExtUtils-Embed hmacalc -y
```

```

1 # 二号陨石坑---缺少一个不可名状的包  rng
2
3 gpg: WARNING: unsafe permissions on homedir `.'
4 gpg: keyring `./secring.gpg' created
5 gpg: keyring `./pubring.gpg' created

```

解决方法

```

1 # 别问，问就是装
2 yum install -y rng-tools-2-13.el6_2.x86_64
3
4 rngd -r /dev/urandom
5
6 # 弄完后重新运行 rpmbuild -bp --target=$(uname -m) kernel.spec
7 # 应该能成功吧 ~ ~ ~ ~

```

开始安装lvs-fullnat

```

1 cd /mnt/
2
3 # 解压前把包放目录里---随便放，主要是用于后边的复制
4 tar zxf Lvs-fullnat-synproxy.tar.gz
5
6 # 回去主安装目录
7 cd~/rpmbuild/BUILD/kernel-2.6.32-220.23.1.el6/linux-2.6.32-220.23.1.el6.x86_64/
8
9 # 从fullnat目录下把补丁拷贝过来
10 cp/mnt/lvs-fullnat-synproxy/lvs-2.6.32-220.23.1.el6.patch . # . 代表当前目录
11
12 # 开始打补丁
13 patch-p1<lvs-2.6.32-220.23.1.el6.patch
14
15 # 修改makefile文件---后面还有很多要修改
16 vi Makefile
17 # 修改这一行:
18 EXTRAVERSION +=-220.23.1.el6
19
20 # 开始配置
21 make

```

```

1 # 三号陨石坑（超多）以下错误无顺序
2
3 #####
4
5 # 1.添加 common.c 的补丁
6 arch/x86/kernel/cpu/common.c: In function 'filter_cpuid_features':

```

```

7 > arch/x86/kernel/cpu/common.c:306: error: implicit declaration of function
8 > 'xen_cpuid_base'
9
10 # 解决方法: 补丁下载链接: ---->打到 common.c 上
11 https://bugzilla.redhat.com/attachment.cgi?id=518670
12
13 #####
14
15 # 2.关于gcc的, 调整gcc的版本
16 gcc: error: elf_i386: No such file or directory
17 make[1]: *** [arch/x86/vdso/vdso32-int80.so.dbg] Error 1
18 make: *** [arch/x86/vdso] Error 2
19
20 # 解决方法: arch/x86/vdso/Makefile 文件中
21 # 28或29行
22 VDSO_LDFLAGS_vdso.lds = -m elf_x86_64 -Wl,-soname=linux-vdso.so.1 \ -Wl,-z,max-page-
23 size=4096 -Wl,-z,common-page-size=4096
24 把"-m elf_x86_64" 替换为 "-m64"
25
26 # 72行左右
27 VDSO_LDFLAGS_vdso32.lds = -m elf_i386 -Wl,-soname=linux-gate.so.1
28 把"-m elf_i386" 替换为 "-m32"
29 #####
30
31 # 3.
32 drivers/scsi/pmraid.h:601:8: error: duplicate member 'sense_buffer'
33
34 # 解决方法 drivers/scsi/pmraid.h
35 //u8 *sense_buffer; # 注释掉这一行
36
37 #####
38
39 # 4.空间不足
40 ld: final link failed: No space left on device
41
42 # 解决方法
43 make mrproper
44
45 #####
46
47 # 5.
48 cc1: some warnings being treated as errors
49
50 # 6.
51 /tmp/ccd1r6iP.s: Assembler messages:
52 /tmp/ccd1r6iP.s: Error: .size expression for do_hypervisor_callback does not evaluate
53 to a constant
54
55 #
56 linux/videodev.h: No such file or directory
57 #解决方法
58 yum install libv4l-dev*

```

```
58 然后将v4lgrab.c中的
59  "#include <linux/videodev.h>"
60 用"#include <libv4l1-videodev.h>"替换
61
62 # dpmap 2.6.32
63 dd if=/dev/zero of=/swap2 bs=1M count=1500
64 mkswap /swap2
65 swapon /swap2
66
67 # 到了这里你就会发现这方法好像错误无穷无尽.....
```

那么我们换一个法子吧（滑稽保命）

```
1 # 先装一下依赖
2 yum install perl perl-devel popt popt-devel libnl libnl-devel libssl libssl-devel
  openssl openssl-devel popt-static
3 yum -y install ncurses ncurses-devel
4 yum install git -y
5 sudo yum -y install gcc gcc-c++ kernel-devel
```

下载源码

```
1 # 源码放置在 usr/local/src/ 目录下了----个人喜好
2 git clone https://github.com/alibaba/LVS.git
```

开始编译内核

```
1 cd /usr/local/src/LVS/kernel
2
3 vi .config
4
5 # 搜索修改这两句话
6 CONFIG_SYSFS_DEPRECATED=n
7 CONFIG_SYSFS_DEPRECATED_V2=n
8
9 # 创建个文件放内核
10 mkdir -p /usr/src/kernels/2.6.32
11
12 # 选择模块需要编译的, exit退出, 选择 YES保存-----换言之: 随便选
13 make O=/usr/src/kernels/2.6.32 menuconfig
14
15 # 清除 .config.....
16 make mrproper
17
18 # 时间很长---可以去恰个饭----遇到错误查看 “三号陨石坑” 之后继续执行
19 make O=/usr/src/kernels/2.6.32
```



```
20
21 # .....
22 make O=/usr/src/kernels/2.6.32 modules_install install
23
```

安装keepalive-----lvs自带

```
1 cd /usr/local/src/LVS/tools/keepalived/
2
3 ./configure --prefix=/usr/local/keepalived --with-kernel-dir=/usr/local/src/LVS-
  lvs_v2/kernel
4
5 make
6
7 make install
```

安装ipvsadm

```
1 cd /usr/local/src/LVS/tools/ipvsadm/
2
3 make
4
5 make install
```

安装quagga

```
1 cd /usr/local/src/LVS/tools/quagga
2
3 ./configure --prefix=/usr/local/quagga
4
5 make
6
7 make install
```

5.dpvs (大坑伤身)

```
1 git clone https://github.com/iqiyi/dpvs.git
```

进入到dpvs目录 安装dpdk , dpvs官方建议 最好把 dpdk放在dpvs目录中

```
1 | https://fast.dpdk.org/re1/dpdk-17.11.2.tar.xz
2 |
3 | tar dpdk-17.11.2.tar.xz
```

注：大坑巨坑

centos可能没有存在正确的yum源，所以需要更新一下（安装某个依赖需要最新的yum源）

```
1 | # 备份一份
2 | mv /etc/yum.repos.d/CentOS-Base.repo /etc/yum.repos.d/CentOS-Base.repo.bar
3 |
4 | #进入下载源目录
5 | cd /etc/yum.repos.d/
6 |
7 | #wget下载----没有wget可以使用rpm或者手动下载拖进去 ((* /w \ *))
8 | wget http://mirrors.163.com/.help/CentOS7-Base-163.repo
9 |
10 | #更新---生成缓存
11 | yum clean all
12 | yum makecache
```

终于可以下载依赖了 注：某些机子可能没有存在正确的源，所以需要上述更新yum源操作

```
1 | yum install numactl-devel
2 |
3 | yum install openssl-devel
4 |
5 | yum install popt-devel
```

编译dpdk

```
1 | cd dpvs
2 |
3 | cp patch/dpdk-stable-17.11.2/*.patch dpdk-stable-17.11.2/
4 |
5 | cd dpdk 路径
6 |
7 | patch -p 1 < 0001-PATCH-kni-use-netlink-event-for-multicast-driver-par.patch
8 |
9 | patch -p1 < 0002-net-support-variable-IP-header-len-for-checksum-API.patch
10 |
11 | make config T=x86_64-native-linuxapp-gcc
12 |
13 | make
14 |
15 | export RTE_SDK=$PWD
```

启动hugepage

```
1 echo 8192 > /sys/devices/system/node/node0/hugepages/hugepages-2048kB/nr_hugepages
2
3 # 这玩意好像没有.....
4 echo 8192 > /sys/devices/system/node/node1/hugepages/hugepages-2048kB/nr_hugepages
5
6 mkdir /mnt/huge
7
8 mount -t hugetlbfs nodev /mnt/huge
9
10 modprobe uio
11
12 cd dpdk-stable-17.11.2
13
14 insmod build/kmod/igb_uio.ko
15
16 insmod build/kmod/rte_kni.ko
```

绑定网卡

查看网卡状态

```
1 ./usertools/dpdk-devbind.py --status
2
3 #发现机子上只有一个网卡怎么办（穷）
4 .....
5 #多配一个
6
7 # Network devices using kernel driver
8
9 # =====
10 # 该字段下面的为网卡，选择没有 * Active * 的
11
12 #绑定网卡，修改的是后面的0000:01:00.2这一段
13 ./usertools/dpdk-devbind.py -b igb_uio 0000:01:00.2
```

编译dpvs

```
1 cd dpdk-stable-17.11.2/
2
3 export RTE_SDK=$PWD
4
5 cd dpvs
6
7 make
8
9 make install
```

启动dpvs # 注意：巨坑来袭

```
1 # 准备修改配置-----至于怎么修改.....
2 cp conf/dpvs.conf.single-nic.sample /etc/dpvs.conf
3
4 cd dpvs/bin
5
6 ./dpvs //
```

查看是否启动

```
1 | ./dpip link show
```