Core description:

The core configuration description is:

- 1C/1T: 1 Physical Core, 1 Logical Core per physical core (1 Hyperthread)

using core #2 (socket 0, 2nd physical core)

- 1C/2T: 1 Physical Core, 2 Logical Cores per physical core (2 Hyperthreads)

using core #2 and #14 (socket 0, 2nd physical core, 2 Hyperthreads)

- 2C/1T: 2 Physical Cores, 1 Logical Core per physical core

using core #2 and #4 (socket 0, 2nd and 3rd physical cores)

--------------------------------

Install DPDK

export http\_proxy=http://proxy-prc.intel.com:911

Clone dpdk:

git clone git:**//**dpdk**.**org**/**dpdk

*# or:*

git clone http:**//**dpdk**.**org**/**git**/**dpdk

git clone <http://dpdk.org/git/dpdk>

for stable release: git clone http://dpdk.org/git/dpdk-stable

for Next tree branch：

git clone http://dpdk.org/git/next/dpdk-next-net

Clone dts:

git clone http://dpdk.org/git/tools/dts

1: configure hugepage

to reserve 4G of hugepage memory in the form of four 1G pages:

default\_hugepagesz=1G hugepagesz=1G hugepages=16

or:

echo 1024 > /sys/devices/system/node/node0/hugepages/hugepages-2048kB/nr\_hugepages

echo 1024 > /sys/devices/system/node/node1/hugepages/hugepages-2048kB/nr\_hugepages

export RTE\_SDK=`pwd`

export RTE\_TARGET=x86\_64-native-linuxapp-gcc

or

       make install RTE\_SDK=`pwd` T=x86\_64-native-linuxapp-gcc -j32

       make -C examples/l3fwd RTE\_SDK=`pwd` T=x86\_64-native-linuxapp-gcc

（gcc 编译用的头文件：kernel-headers,kernel-devel，文件后需带内核版本号）

mkdir /mnt/huge

mount -t hugetlbfs nodev /mnt/huge

make install T=x86\_64-native-linuxapp-gcc

/bin/sh: 2: cc: not found apt install gcc

# yum install numactl-devel when encounter fatal error “numa.h: No such file or directory” when compiling dpdk

If the same problem for Ubuntu OS, please: apt-get install libnuma-dev to fix it.

* BIOS upgrade tool:

Dedi prog SF100 ISF

To download driver and software:

<http://www.dediprog.com/pd/spi-flash-solution/SF100>

* configure DTS:
* git clone http://dpdk.org/git/tools/dts

vim execution.cfg

vim conf/crbs.cfg

vim conf/ports.cfg

lspci –nn | grep Eth

ethtool –I ethname

set proxy:

export http\_proxy=http://proxy-prc.intel.com:911

export https\_proxy=https://proxy-prc.intel.com:911

proxy=http://proxy-prc.intel.com:911 (for fedora dnf usage, append this content to /etc/dnf/dnf.conf)0

(Ubuntu17.04 在~/.profile 设置可永久生效, 其他版本可能在~/.bashrc里面设置)

* file share server:

IP: [\\10.240.176.131](file:///\\10.240.176.131)

* update BIOS:(not fanalized)

1. advance🡪process configuration🡪Turbo🡪Disable🡪F10
2. reset
3. F2🡪boot manager

Git log

Git apply \*.path

Git reset –hard HEAD

Dhclient – ethname

# 总核数 = 物理CPU个数 X 每颗物理CPU的核数

# 总逻辑CPU数 = 物理CPU个数 X 每颗物理CPU的核数 X 超线程数

# 查看物理CPU个数

cat /proc/cpuinfo| grep "physical id"| sort| uniq| wc -l

# 查看每个物理CPU中core的个数(即核数)

cat /proc/cpuinfo| grep "cpu cores"| uniq

# 查看逻辑CPU的个数

cat /proc/cpuinfo| grep "processor"| wc -l

查看CPU信息（型号）

cat /proc/cpuinfo | grep name | cut -f2 -d: | uniq -c

查看内 存信息

# cat /proc/meminfo

Redhat 7.3 install scapy：

Download scapy：

Wget http://mirrors.aliyun.com/fedora/releases/24/Everything/x86\_64/os/Packages/s/scapy-2.3.1-2.fc24.noarch.rpm

rpm -ivh scapy-2.3.1-2.fc24.noarch.rpm

tar -czf small.tar.gz small(目录名)  ;压缩并打包目录。

[‎2/‎6/‎2017 9:44 AM] Lu, PeipeiX:

mount -o remount,rw /

git usage:

git apply \*.patch

scp 10.67.118.31:/home/osimg/ubuntu16.img

git clone http://dpdk.org/git/dpdk

然后

git tag

再git checkout

指定的版本

export http\_proxy=http://proxy-prc.intel.com:911

编译pktgen,先编译dpdk，再编译它

在pktgen 目录：

export RTE\_SDK=dpdk目录

export RTE\_TARGET=x86\_64-native-linuxapp-gcc

安装patch: dnf install patch

dnf install readline-devel –y

在tester上运行testpmd

CONFIG\_RTE\_LIBRTE\_MLX4\_PMD=y

make -j 4 install T=x86\_64-native-linuxapp-gcc

pktgen usage：

to get pktgen source code:

git clone http://dpdk.org/git/apps/pktgen-dpdk

1：edit the /etc/sysctl.conf file to setup the hugepages size:

$ sudo vi /etc/sysctl.conf

Add to the bottom of the file:

vm.nr\_hugepages=256

2：edit the /etc/fstab file to mount the hugepages at startup:

$ sudo vi /etc/fstab

Add to the bottom of the file:

huge /mnt/huge hugetlbfs defaults 0 0

$ sudo mkdir /mnt/huge

$ sudo chmod 777 /mnt/huge

Show bios version

dmidecode -t bios

perf top

dmesg

cat /proc/cpuinfo

cat /proc/cmdline

vim /etc/default/grub

intel\_iommu=on iommu=pt

update-grub (activate the new config for Ubuntu arch OS)

grub2-mkconfig (activate the new config for fedora arch OS)

how to open telnet function for Ubuntu for remote login

<http://blog.csdn.net/ycc541/article/details/46610311>

<http://www.cnblogs.com/junsky/archive/2009/08/01/1536361.html>

[Fedora开启telnet服务](http://blog.csdn.net/fm0517/article/details/39183019)

安装telnet服务：  
#yum install telnet-server xinetd -y  
  
重启telnet服务：  
#systemctl enable telnet.socket  
#systemctl start telnet.socket  
  
关闭防火墙：

ufw disable/enable

安装相关库：

install python-xlwt

install python-pexpect

install python-numpy

install python-docutils

install python-pcapy

install scapy

压缩dpdk到dts/dep下面

配置conf/crbf.cfg ports.cfg

Ip link show

把execution/execution.cfg 复制到dts文件夹下

Qemu -serial telnet:localhost:5432

需要在VM里面修改： /boot/grub2/grub.cfg

linux16 /vmlinuz-4.4.3-300.fc23.x86\_64 root= 一行末尾添加 console=ttyS0,115200

mputty,putty, mobaXterm

performance test report

<http://fast.dpdk.org/doc/perf/DPDK_17_02_Intel_NIC_performance_report.pdf>

weekly report

要  cc   to  NPG-PRC-SW STV [npg-prc-sw.stv@intel.com](mailto:npg-prc-sw.stv@intel.com)

; NPG-PRC-SW STV CW [npg-prc-sw.stv.cw@intel.com](mailto:npg-prc-sw.stv.cw@intel.com)

17.02 performance test report

<http://fast.dpdk.org/doc/perf/DPDK_17_02_Intel_NIC_performance_report.pdf>

hugepage operate

cat /proc/meminfo |grep -i huge

rm -fr /mnt/huge/\*

ls /dev/hugepages/\*

rm -fr /dev/hugepages/\*

mount |grep -i huge

git 查询提交者：

git log --pretty=format:"%h - %an, %ar : %s"

for more about how to use git:

<https://git-scm.com/book/zh/v1/Git-%E5%9F%BA%E7%A1%80-%E6%9F%A5%E7%9C%8B%E6%8F%90%E4%BA%A4%E5%8E%86%E5%8F%B2>

**ethtool -s <dev> autoneg on**

apply leave email:

[bg1\_intel\_ops@pactera.com](mailto:bg1_intel_ops@pactera.com)

pip install  xxxxxx

有关多核cpu的linux命令： top 按 1 或者htop 或用tasket

看亲核性： cat/proc/[pid]/task/[tid]/status

Testpmd –nb-cores =N 设置N个core做转发。不设默认都是一个core转发，即使 –c 或-l参数设置了多个core。

Start a second testpmd:

./x86\_64-native-linuxapp-gcc/app/testpmd -c 0x0f -n 4  -b 0000:0a:00.0 --file-prefix=vhost1 --socket-mem=512,0 -- -i

git config --global http.proxy http://proxy-prc.intel.com:911

re

验证0.25倍数的正则表达式：  /^\d+(\.(0|25|5|75)0\*)?$/g

验证非负整数： /^(0|[1-9][0-9]\*)$/

验证正整数：/^([1-9][0-9]\*)$/

验证浮点数：/^(0(\.\d\*[1-9]+\d\*)?)$|^([1-9]\d\*)(\.\d\*)?$/

Mellanox official performance test report

<http://fast.dpdk.org/doc/perf/DPDK_17_05_Mellanox_NIC_performance_report.pdf>

intel nic driver download：

<https://sourceforge.net/projects/e1000/>

[fengqin.wang@pactera.com](mailto:fengqin.wang@pactera.com)

[qiong.wu7@pactera.com](mailto:qiong.wu7@pactera.com)

[win10 .net framework 3.5无法安装错误代码0x800F081F](http://blog.csdn.net/kr0920/article/details/70153935)

<http://blog.csdn.net/kr0920/article/details/70153935>

新装 Ubuntu 安装并打开sshd的步骤：

<http://blog.csdn.net/weiwei_pig/article/details/50954334>

ixia automation expert:

[leyu@ixiacom.com](mailto:leyu@ixiacom.com)

ftp://10.67.118.21/d

start VM

qemu-system-x86\_64 -enable-kvm -m 4096 -smp cores=2,sockets=1 -cpu host -name dpdk1-f23 -drive file=/home/osimg/f23.img -netdev tap,id=ipvm1,ifname=tap3,script=/etc/qemu-ifup -device e1000,netdev=ipvm1,id=net0,mac=00:00:00:00:00:01 -localtime -vnc :10 -daemonize

apt-get install libnuma-dev

vhost-user : ./testpmd -n 4 -c 0x0c --socket-mem 1024 --no-pci --file-prefix=virtio --vdev=net\_virtio\_user0,mac=00:01:02:03:04:05,path=./vhost-net -- -i --txqflags=0xf00 --disable-hw-vlan-filter --port-topology=chained

dut : /testpmd -n 4 -c 0x03 --socket-mem 1024--file-prefix=vhost --vdev 'net\_vhost0,iface=vhost-net,queues=1,client=0' -- -i --port-topology=chained

ftp://10.67.118.21/d

export RTE\_SDK=`pwd`

export RTE\_TARGET=x86\_64-native-linuxapp-gcc

make –j 30 install T=x86\_64-native-linuxapp-gcc

export http\_proxy=<http://proxy-prc.intel.com:911>

How to do the image

<http://wiki.ir.intel.com/sw/index.php/DPDK:_Fedora-23_installation_in_VM_image>

Some specific setting in Ubuntu

<http://wiki.ir.intel.com/sw/index.php/DPDK:_Ubuntu-12.04_installation_in_VM_image>

ERROR: Test Case test\_perf\_virtio\_modern\_qemu\_vector\_pmd Result FAILED: TIMEOUT on x86\_64-native-linuxapp-gcc/app/testpmd -c 0x3 -n 3 -- -i --txqflags=0xf01 --disable-hw-vlan-filter

挂载一个模块

modprobe uio

安装驱动

insmod x86\_64-native-linuxapp-gcc/kmod/igb\_uio.ko

绑定驱动

usertools/dpdk-devbind.py -s |less

./usertools/dpdk-devbind.py -b igb\_uio 03:00.0

每次开机运行：

mount -t hugetlbfs nodev /mnt/huge

modprobe uio

insmod x86\_64-native-linuxapp-gcc/kmod/igb\_uio.ko

./usertools/dpdk-devbind.py -b igb\_uio 00:03.0

解绑网卡：

./usertools/dpdk-devbind.py -u 0000:83:00.0

第三周：

1. test case
2. 安装和编译qemu 2.6, 2.7, 2.8, 2.9, 2.10

 ./configure  --target-list=x86\_64-softmmu --prefix=/home/qemu /qemu\_2.10

./configure --target-list=x86\_64-softmmu --prefix=/home/tester/software/qemu-2.10.0/

make -j 5

make install

手动测试下Multi queue的case

用Ubuntu default的qemu

3、那你在这台机器上准备好DPDK的编译环境，DTS的运行环境，和各种qemu 版本

遇到Unable to locate package autoreconf问题，应该：sudo apt-get install autoconf automake libtool

flex: Command not found问题：yum install -y flex

[bison: Command not found](http://blog.csdn.net/weborn/article/details/6794671) 问题：yum  install -y   bison  bison-devel

default\_hugepagesz=1G hugepagesz=1G hugepages=16

tar –xvf xx.tar.xz

1、\*.tar 用 tar –xvf 解压

　　2、\*.gz 用 gzip -d或者gunzip 解压

　　3、\*.tar.gz和\*.tgz 用 tar –xzf 解压

　　4、\*.bz2 用 bzip2 -d或者用bunzip2 解压

　　5、\*.tar.bz2用tar –xjf 解压

　　6、\*.Z 用 uncompress 解压

　　7、\*.tar.Z 用tar –xZf 解压

　　8、\*.rar 用 unrar e解压

　　9、\*.zip 用 unzip 解压

查看相应版本

apt-cache search glib2

tar –czf jpg.tar.gz \*.jpg   //将目录里所有jpg文件打包成jpg.tar后，并且将其用gzip压缩，生成一个gzip压缩过的包，命名为jpg.tar.gz

1、try run Vhost/Virtio Basic rx/tx, Vhost/Virtio-net Basic rx/tx and Vhost/Virtio multiple queue test cases manually

2、make install qemu v2.6, v2.7, v2.8, v2.9,v 2.10

3、install ubuntu16.10 ,make install dpdk, run dts and install qemu v2.6, v2.7,v 2.8, v2.9, v2.10 on the other server

4、

scp 10.67.118.31:/home/lei/

git clone git@10.67.118.31:/home/lei/dts\_git/dts

vm\_power\_manager:

Makefile:33: \*\*\* vm\_power\_manager requires libvirt >= 0.9.3. Stop.

Apt install libvirt-dev

1、halt   立刻关机 2、poweroff  立刻关机 3、shutdown -h now 立刻关机(root用户使用) 4、shutdown -h 10 10分钟后自动关机 如果是通过shutdown命令设置关机的话，可以用shutdown -c命令取消重启

1、reboot 2、shutdown -r now 立刻重启(root用户使用)

<https://gia.intel.com/guest/create_user_receipt.php>

sh /home/lei/dpdk/Guest\_script.sh

VIRTDUT\_CMD: ifconfig ens4 1.1.1.2

VIRTDUT\_CMD: ifconfig ens4 up

VIRTDUT\_CMD: arp -s 1.1.1.3 52:54:00:00:00:02

VIRTDUT\_CMD: iperf -s -p 12345 -i 1 > iperf\_server.log &

VIRTDUT\_CMD: ifconfig -a

VIRTDUT\_CMD: sh /home/lei/dpdk/Guest\_script.sh

VIRTDUT\_CMD: ifconfig ens4 1.1.1.3

VIRTDUT\_CMD: ifconfig ens4 up

VIRTDUT\_CMD: arp -s 1.1.1.2 52:54:00:00:00:01

VIRTDUT\_CMD: iperf -c 1.1.1.2 -p 12345 -i 1 -t 60 > iperf\_client.log &

export RTE\_SDK=`pwd`

export RTE\_TARGET=x86\_64-native-linuxapp-gcc

make -j 30 install T=x86\_64-native-linuxapp-gcc

注意：测试时，dpdk版本要一致

ISMP

https://e2esm.intel.com/nav\_to.do?uri=%2Fhome\_splash.do%3Fsysparm\_direct%3Dtrue

https://securityportal.intel.com/AccessProvisioningModule/

enp177s0f1: flags=4419<UP,BROADCAST,RUNNING,PROMISC,MULTICAST> mtu 1500

inet 1.1.1.8 netmask 255.0.0.0 broadcast 1.255.255.255

inet6 fe80::92e2:baff:fed5:b065 prefixlen 64 scopeid 0x20<link>

ether 90:e2:ba:d5:b0:65 txqueuelen 1000 (Ethernet)

RX packets 9857572 bytes 652348610 (652.3 MB)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 55475388 bytes 83893112460 (83.8 GB)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp177s0f1: flags=4419<UP,BROADCAST,RUNNING,PROMISC,MULTICAST> mtu 1500

inet 1.1.1.8 netmask 255.0.0.0 broadcast 1.255.255.255

inet6 fe80::92e2:baff:fed5:b065 prefixlen 64 scopeid 0x20<link>

ether 90:e2:ba:d5:b0:65 txqueuelen 1000 (Ethernet)

RX packets 14309136 bytes 946578934 (946.5 MB)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 64362541 bytes 97348154042 (97.3 GB)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

tap0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

inet 1.1.1.2 netmask 255.0.0.0 broadcast 1.255.255.255

inet6 fe80::3cdd:fcff:fe3d:5e22 prefixlen 64 scopeid 0x20<link>

ether 3e:dd:fc:3d:5e:22 txqueuelen 1000 (Ethernet)

RX packets 0 bytes 0 (0.0 B)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 39 bytes 2306 (2.3 KB)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

打补丁：

git add tests/TestSuite\_vhost\_port\_start\_stop.py

git commit -m "vhost port start stop patch"

git format-patch -X --signoff --subject-prefix=`PATCH VX`

ifconfig ens4 up

ifconfig ens4 1.1.1.2

ifconfig ens4 up

arp -s 1.1.1.3 52:54:00:00:00:02

iperf -s -p 12345 -i 1

ifconfig ens4 1.1.1.3

ifconfig ens4 up

arp -s 1.1.1.2 52:54:00:00:00:01

iperf -c 1.1.1.2 -p 12345 -i 1 -t 5

<http://www.study-area.org/compu/compu_mem.htm>

<http://www.study-area.org/network/network.htm>

vhost\_loopback\_performance\_virtio\_user v17.08,v17.11-rc1,v17.11-rc2