**TÍCH HỢP OVS-DPDK VỚI OPENSTACK**

1. **Tích hợp OVS-DPDK với OpenStack**
   1. **Chuẩn bị**
      1. **Kiểm tra phần cứng**

* Kích hoạt tính CPU hyperthreading và kiểm tra lại:

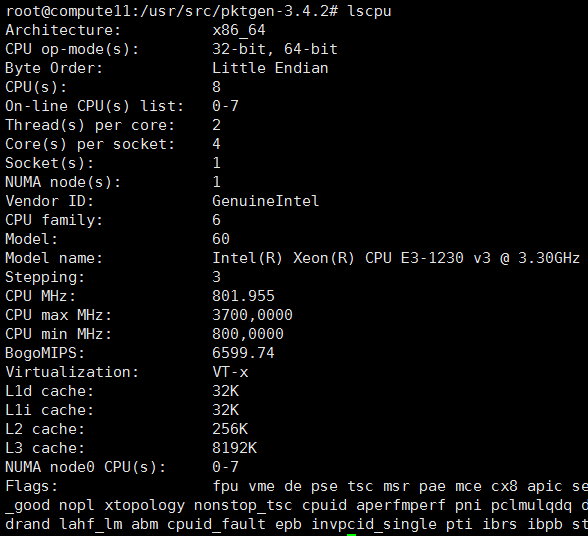
|  |
| --- |
| dmidecode -t processor | grep HTT |



* Kiểm tra thông tin CPU và NUMA:

|  |
| --- |
| lscpu |

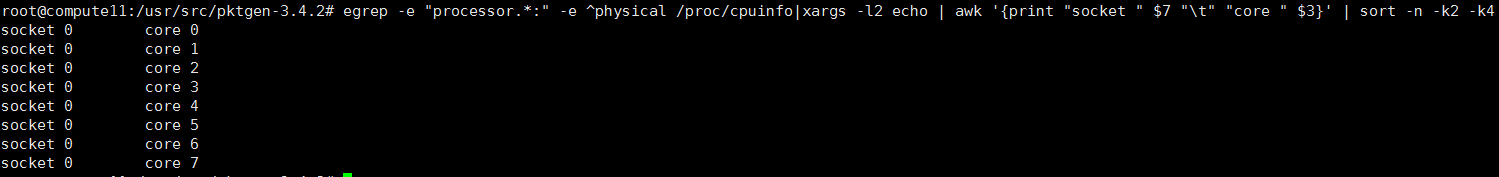
* Kết quả tương tự như sau:



* Thông tin cần thiết:
* Số lượng NUMA node: 1
* Số lượng sockets: 1
* Số lượng core/socket: 4
* Số threads/core: 2
* Tổng số CPU: 1 x 4 x2 = 8 (CPU IDs: 0-7)
* Xem topology về CPU
* Mapping giữa CPU và NUMA node chứa các CPU:

|  |
| --- |
| egrep -e "processor.\*:" -e ^physical /proc/cpuinfo|xargs -l2 echo | awk '{print "socket " $7 "\t" "core " $3}' | sort -n -k2 -k4 |

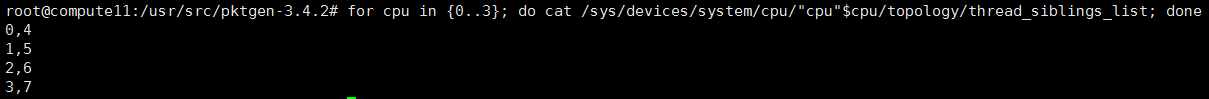
Kết quả:



* Kiểm tra các cặp hyperthreads trên mỗi core, ví dụ server có tổng cộng 4 cores:

|  |
| --- |
| for cpu in {0..3}; do cat /sys/devices/system/cpu/"cpu"$cpu/topology/thread\_siblings\_list; done |

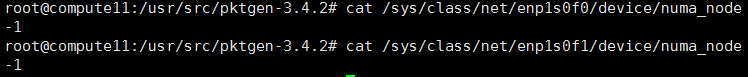
Kết quả:



* Kiểm tra xem các DPDK NIC (10GbE NIC) map với NUMA node nào. Ví dụ:

|  |
| --- |
| cat /sys/class/net/enp1s0f0/device/numa\_node  cat /sys/class/net/enp1s0f1/device/numa\_node |

Kết quả tương tự như sau:



* + 1. **Tính toán các tham số chuẩn bị cho OVS-DPDK**
* CPU Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tham số | Ý nghĩa | Giá trị |
| 1 | OVS\_PMD\_CORE\_MASK | Danh sách các CPU chuyển sang hệ hexa. Sử dụng để cấp cho các PMD threads của Open vSwitch (tham số quan trọng ảnh hưởng hiệu năng) | Ví dụ sử dụng 4 CPU: 2,3,6,7. Chuyển sang hexa: 0xcc |
| 2 | OVS\_CORE\_MASK | Danh sách các CPU chuyển sang hệ hexa. Sư dụng cho **dpdk lcore threads** | Tương tự OVS\_PMD\_CORE\_MASK. Ví dụ: dùng core 0 🡪 0x1 |
| 3 | Nova CPUs pin set | Danh sách các CPU sử dụng cấp phát cho các OpenStack instances | Phải khác với hai tham số (1) và (2). Ví dụ: dùng core 1,4,5 🡪 0x32 |
| 4 | Host isosolated core list | Là tập hợp các CPU core sử dụng cho OVS-DPDK và các OpenStack instance | Là danh sách toàn bộ các CPU (1), (2), (3) tập hợp lại. Giá trị này dùng để cấu hình cho hệ điều hành khi khởi động, các tiến trình của hệ điều hành sẽ không sử dụng các core nằm trong danh sách này. |

***Chú ý:*** Với những server có nhiều CPU, để tính toán CPU mask nhanh hơn, có thể sử dụng python command tương tự như sau:

|  |
| --- |
| python  >> print "%x" % ((1<< 10) | (1<< 11) | (1<< 34) | (1<< 35))  c00000c00 |

Trong đó: ***10, 11, 34, 35*** là danh sách các CPU.

* Memory Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tham số | Ý nghĩa | Giá trị |
| 1 | DPDK number of memory channels | Số lượng memory channels trên mỗi NUMA node | * Xác định số memory channels:   **dmidecode -t memory**   * Xác định số numa nodes:   ls /**sys/devices/system/node/node\* -d**   * DPDK memory channels = **(memory channels / số numa nodes)** |
| 2 | OVS\_NUM\_HUGEPAGES | Tổng số Hugepages sử dụng cho OVS-DPDK và các VMs | Tổng số Huge pages (kích thước 2MB). Giả sử dùng 6 GB 🡪 Tổng số Hupages: 6 x 1024/2 = 3072 pages |
| 3 | OVS\_SOCKET\_MEM | Socket memory sử dụng cho OVS-DPDK | * Tính theo MTU mặc định trên hệ thống Cloud. Ví dụ với MTU = 1500. Các bước tính như sau: * Làm tròn lên cho MTU tới bội của 1024: 1500 🡪 2048 * Socket memory tính theo công thức: * MEMORY\_REQD\_PER\_MTU = (ROUNDUP\_PER\_MTU + 800) \* (4096 \* 64)   🡪 (2048 + 800) \* (4096 \*64) = 746586112 (bytes) = 712 MB 🡪 Làm tròn lên bội gần nhất của 1024 🡪 Socket memory: 1024 MB   * Nếu hệ thống Cloud sử dụng nhiều giá trị MTU khác nhau (thông thường là jumbo frames 9000 và 2000) 🡪 tính giá trị memory với từng MTU và lấy tổng các giá trị này (MB) 🡪 Làm tròn lên tới bội gần nhất của 1024. Ví dụ hệ thống Cloud sử dụng 2 loại MTU là: 9000 và 2000 🡪 Socket mem: 4096 MB. * Với Socket chứa DPDK NIC, giá trị socket mem được tính theo công thức đưa ra phía trên. Với socket không chứa DPDK NIC, giá trị này để mặc định là: 1024. 🡪 ví dụ với server có 2 socket: socket 0 chứa DPDK NIC, socket 1 không chứa DPDK NIC   🡪 OVS\_SOCKET\_MEM=4096,1024 (sử dụng dấu phẩy ngăn cách giá trị socket mem giữa các socket) |
| 4 | Reserved Host Memory | Dung lượng bộ nhớ dùng cho các tiến trình thông thường của hệ điều hành | Tổng bộ nhớ hệ thống – OVS\_NUM\_HUGEPAGES x 2 (MB). Ví dụ: 8192 – 3072 x 2 = 2048 (MB) |

* Networking Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tham số | Ý nghĩa | Giá trị |
| 1 | DPDK driver type  (OVS\_INTERFACE\_DRIVER) | Driver type sử dụng bởi DPDK | ***vfio-pci*** hoặc ***igb\_uio*** |
| 2 | Neutron datapath type | Datapath type của OVS bridge | ***netdev*** |
| 3 | vhostuser socket directory (OVS\_VHOST\_USER\_SOCKET\_DIR) | Thư mục chứa vhost-user của OVS | Tùy chỉnh, mặc định là: ***/run/openvswitch*** |

* Các tham số khác

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tham số | Ý nghĩa | Giá trị |
| 1 | nova-scheduler default filters |  | Tham khảo phần cấu hình ở mục 1.2.3 |
| 2 | Compute kernel parameters | Các tham số cấu hình trong /etc/default/grub: hugepagesz (2MB hoặc 1 GB), hugepages, iommu, isolcpus | * Cấu hình mẫu trong **/etc/default/grub**:   **GRUB\_CMDLINE\_LINUX\_DEFAULT="in default\_hugepagesz=2MB hugepagesz=2MB intel\_iommu=on iommu=pt isolcpus=3-10"** |

* 1. **Các bước thực hiện**
     1. **Backup dự phòng**
* Trong trường hợp Cloud còn đủ tài nguyên, live migrate hoặc migrate các VMs trên Compute Node cần nâng cấp từ OVS 🡪 OVS-DPDK sang một Compute Node khác. Sử dụng command:

|  |
| --- |
| openstack server migrate <VM\_ID> --live <TARGET\_COMPUTE> |

* Trong trường hợp Cloud không còn đủ tài nguyên, tắt toàn bộ các VMs trên các Compute Node cần nâng cấp OVS 🡪 OVS-DPDK.
* Chú ý:
* Tắt dịch vụ ***masakari-hostmonitor*** trên các Compute Node được cấu hình cùng Masakari segment với Compute Node hiện tại. Việc này tránh tình huống các Compute Node cùng Masakari segment với Compute Node hiện tại gọi API cứu các VMs trên Compute này.
* Xóa ***neutron-dhcp-agent*** trên Compute Node (nếu có):
* Lấy danh sách các neutron agent:

|  |
| --- |
| neutron agent-list |

* Lấy DHCP agent ID trên Compute Node hiện tại và xóa agent này:

|  |
| --- |
| neutron agent-delete <dhcp\_agent\_id> |

* Nếu Compute Node đang chứa các network namespace của OpenStack (dhcp, Floating IP, router namespaces), trước khi cài đặt OVS-DPDK, ngắt hẳn kết nối OpenStack Management và xóa toàn bộ các network namespace hiện có:

|  |
| --- |
| ifdown <management\_interface>  ip --all netns delete |

Trong đó ***<management\_interface>*** là NIC trên dải management APIs của OpenStack. Bằng cách này, các namespace (ví dụ DHCP namespace) sẽ tự động được di chuyển sang các Compute Node khác, tránh ảnh hưởng tới các VMs trên hệ thống trong quá trình nâng cấp.

* Disable dịch vụ nova-compute trên Compute Node hiện tại. Trên controller node bất kì, thực thi command sau:

|  |
| --- |
| openstack compute service set --disable <COMPUTE\_NODE> nova-compute |

* + 1. **Cài đặt OVS-DPDK**
* Tải scripts cài đặt OVS-DPDK từ gitlab: [***https://gitlab.visc.com/thaiph5/openvswitch/***](https://gitlab.visc.com/thaiph5/openvswitch/) hoặc giải nén scripts đính kèm tài liệu:

|  |
| --- |
| tar xzvf ovs-dpdk.tar.gz |

* Chỉnh sửa lại các tham số dùng để cài đặt OVS-DPDK: Khai báo các tham số sau trong tệp: ***ovs-dpdk/settings***

|  |  |
| --- | --- |
| **Tham số** | **Giá trị** |
| OVS\_DPDK\_GIT\_TAG | DPDK branch sử dụng, mặc định: v17.05.1 |
| OVS\_GIT\_TAG | Open vSwitch branch, mặc định: v2.8.0 |
| OVS\_HUGEPAGE\_MOUNT | Thư mục mount hugepages (/mnt/huge) |
| OVS\_HUGEPAGE\_MOUNT\_PAGESIZE | Kích thước hupages (2MB | 1GB) |
| OVS\_NUM\_HUGEPAGES | Số lượng hugepages (tham số memory số (2) trong mục 1.1.2) |
| OVS\_SOCKET\_MEM | Tham số memory số (3) trong mục 1.1.2 |
| OVS\_MEM\_CHANNELS | Tham số memory số (1) trong mục 1.1.2 |
| OVS\_CORE\_MASK | Tham số cpu số (2) trong mục 1.1.2 |
| OVS\_PMD\_CORE\_MASK | Tham số cpu số (1) trong mục 1.1.2 |
| OVS\_INTERFACE\_DRIVER | DPDK NIC driver, tham số networking số (1) trong mục 1.1.2 |
| OVS\_BRIDGE\_MAPPINGS | Mapping giữa OpenStack network và OVS-DPDK bridge |
| OVS\_PCI\_MAPPINGS | 10 GbE NIC |
| OVS\_VHOST\_USER\_SOCKET\_DIR | Tham số networking số (3) trong mục 1.1.2 |

* Update, upgrade hệ điều hành và update grub:

|  |
| --- |
| apt-get update && apt-get dist-upgrade –y |

Update grub để cấu hình các isolated CPUs dùng cho các VMs và Open vSwitch, chỉnh sửa trong file ***/etc/default/grub***:

|  |
| --- |
| **GRUB\_CMDLINE\_LINUX\_DEFAULT="in default\_hugepagesz=2M hugepagesz=2M intel\_iommu=on iommu=pt isolcpus=3-10"** |

Trong đó:

* **default\_hugepagesz, hugepagesz**: kích thước hugepages thiết lập cho hệ điều hành. Giá trị: 2M hoặc 1G
* **isolcpus:** danh sách các CPUs cấp cho VMs và OVS-DPDK

Lưu lại tệp này, cập nhật grub và reboot: ***update-grub && init 6***

* Cài đặt OVS-DPDK:

|  |
| --- |
| cd ovs-dpdk;  chmod +x setup.sh;  bash setup.sh stack pre-install;  bash setup.sh stack install; |

* Nếu gặp trục trặc trong quá trình cài đặt, clean dữ liệu của OVS-DPDK:

|  |
| --- |
| bash setup.sh unstack  bash setup.sh clean |

* Sau khi cài đặt, sử dụng một số command cơ bản của OVS-DPDK sau:

|  |
| --- |
| # khởi động ovs-dpdk  /etc/init.d/ovs-dpdk start  # tắt ovs-dpdk  /etc/init.d/ovs-dpdk stop  # restart ovs-dpdk  /etc/init.d/ovs-dpdk restart |

* + 1. **Cấu hình sau cài đặt**
* Cấu hình Nova:
* Trên tất cả các OpenStack Nodes: mở file ***nova.conf*** và cấu hình như sau:

|  |
| --- |
| [DEFAULT]  …  scheduler\_default\_filters = RamFilter,ComputeFilter,AvailabilityZoneFilter,ComputeCapabilitiesFilter,ImagePropertiesFilter,PciPassthroughFilter,NUMATopologyFilter  allow\_resize\_to\_same\_host = True  [scheduler]  driver = filter\_scheduler  [filter\_scheduler]  enabled\_filters = RetryFilter,AvailabilityZoneFilter,RamFilter,DiskFilter,ComputeFilter,ComputeCapabilitiesFilter,ImagePropertiesFilter,ServerGroupAntiAffinityFilter,ServerGroupAffinityFilter,SameHostFilter,Dif  ferentHostFilter |

* Trên Compute Node: mở file ***nova.conf*** và cấu hình dưới section [DEFAULT]:

|  |
| --- |
| [DEFAULT]  …  vcpu\_pin\_set = 2,3,4,6,7 |

Trong đó: **2,3,4,6,7** là tham số **(3) Nova CPUs pin set** liệt kê ở mục ***1.1.2.***

Restart lại các dịch vụ trên các OpenStack Node:

* Controller Nodes:

|  |
| --- |
| docker restart nova-placement-api  docker restart nova-novncproxy  docker restart nova-conductor  docker restart nova-scheduler  docker restart nova-consoleauth  docker restart nova-api |

* Compute Nodes:

|  |
| --- |
| systemctl retstart libvirtd  systemctl restart nova-compute |

* Enable lại dịch vụ nova-compute trên các Compute Node đã cài OVS-DPDK:

|  |
| --- |
| openstack compute service set --enable compute11 nova-compute |

* Tạo Nova Aggregation: Để tạo mới các instance tận dụng hiệu năng của OVS-DPDK, tạo Nova Aggregation tập hợp các Compute Host đã cài đặt OVS-DPDK như sau:

|  |
| --- |
| # create AZ & Aggregation  nova aggregate-create HIGH-PERF-NET HIGH-PERF-NET  # add compute host to aggregation  nova aggregate-add-host HIGH-PERF-NET compute11 |

Lần lượt bổ sung các Compute Host khác vào aggregation đã tạo. Trong quá trình launch các VMs sau này, chú ý chọn Availability Zone (Aggregation) chứa các Compute Host đã cài OVS-DPDK.

* Cấu hình Neutron: Thực hiện trên các Compute Node:
* Load docker image đính kèm (đây là docker image mới đã cài đặt sẵn neutron-dpdk plugin):

|  |
| --- |
| docker load < neutron-dpdk.tar |

* Chỉnh sửa lại tệp ***openvswitch\_agent.ini***:

|  |
| --- |
| vim /usr/share/docker/neutron/neutron/plugins/ml2/openvswitch\_agent.ini |

Nội dung tệp này tương tự như sau:

|  |
| --- |
| [DEFAULT]  [agent]  tunnel\_types = vxlan  l2\_population = True  [ovs]  bridge\_mappings = net0:br-enp1s0f0,net1:br-enp1s0f1,mgmt:br-mgmt  local\_ip = 192.168.2.102  vhostuser\_socket\_dir = /run/openvswitch/  datapath\_type = netdev  tunnel\_bridge = br-tun  [securitygroup]  firewall\_driver = iptables\_hybrid  [xenapi] |

Trong đó, chú ý chỉnh sửa lại các tham số dưới section ***[ovs],*** chú ý hai tham số quan trọng:

* ***vhostuser\_socket\_dir***: thư mục chứa các vhost-user socket, giá trị mặc định /run/openvswitch
* ***datapath\_type:*** datapath type khi tạo mới openvswitch instance (bridge, port,…), giá trị mặc định ***netdev (hoặc system với bản Open vSwitch thông thường)***
* Tạo các openvswitch bridge tương ứng với cấu hình trong tệp ***openvswitch\_agent.ini***:

|  |
| --- |
| ovs-vsctl add-br br-enp1s0f0 -- set bridge br-enp1s0f0 datapath\_type=netdev  ovs-vsctl add-br br-enp1s0f1 -- set bridge br-enp1s0f1 datapath\_type=netdev  ovs-vsctl add-br br-mgmt -- set bridge br-mgmt datapath\_type=netdev |

Add các physical interface vào các bridge theo nhu cầu để đảm bảo kết nối cho các VMs.

* Xóa ***neutron-ovs-agent*** cũ đi và tạo mới OpenvSwitch agent từ docker image mới:
* Tạo tệp chứa biến môi trường dùng để run Open vSwitch agent container:

|  |
| --- |
| cat << EOF > /usr/share/docker/neutron/neutron/neutron.env  export MEMCACHE\_SECRET\_KEY=CFljCXQxb5Q97KYExEojZaEPbEOMaYrr  export http\_proxy=  export REGION\_NAME=North\_VN  export MY\_IP=192.168.2.102  export TRANSPORT\_URL=rabbit://openstack:2SHdjx6ZW5IOIUQdzY72eVnt8SYuV2So@controller01:5672,openstack:2SHdjx6ZW5IOIUQdzY72eVnt8SYuV2So@controller02:5672,openstack:2SHdjx6ZW5IOIUQdzY72eVnt8SYuV2So@controller03:5672  export NOVA\_PASS=GUWDgSMPhpCQfshTxqqh6Hz1RgjNNPd0  export CELL\_NAME=cell1  export METADATA\_PROXY\_SHARED\_SECRET=vmpvE90N1rAZqlN703GGyC6IXz9yXMRS  export https\_proxy=  export NEUTRON\_PASS=LsmNqNppRxEVqHVKhVp7wsMaCHw7qnn0  export REGISTRY\_HOST=os-controller  export DB\_HOST=os-controller  export NEUTRON\_DBPASS=3FgxfunYkARQvcc8J5NI2Gybn8cVer7U  export MEMCACHED\_SERVER=192.168.2.102  export RABBIT\_HOSTS=controller01:5672,controller02:5672,controller03:5672  export MEMCACHED\_SERVERS=controller01:11211,controller02:11211,controller03:11211  export CONTROLLER\_HOST=os-controller  export TUNNEL\_IP=192.168.2.102  export NEUTRON\_START=START\_NEUTRON\_OVS\_AGENT  export SHARED\_CONF\_DIR=/usr/share/docker  export PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin  export DEBIAN\_FRONTEND=noninteractive  export SERVICE=neutron  export NEUTRON\_OVS\_DPDK=http://github.com/openstack/networking-ovs-dpdk.git  export NEUTRON\_OVS\_DPDK\_BRANCH=master"  EOF |

* Chỉnh sửa lại tệp biến môi trường này cho phù hợp và run neutron ovs agent:

|  |
| --- |
| docker rm –f –v neutron-ovs-agent  docker run -d --restart=unless-stopped --name neutron-ovs-agent --network=host --privileged --user neutron \  -v /var/log/neutron:/var/log/neutron \  -v /var/lib/neutron:/var/lib/neutron \  -v /run:/run:shared \  -v /run/netns:/run/netns:shared \  -v /run/openvswitch:/run/openvswitch \  -v /etc/localtime:/etc/localtime \  --env-file=/usr/share/docker/neutron/neutron/neutron.env \  docker-registry/neutron-dpdk:pike |

* Chú ý: nếu cần thiết tạo lại neutron-dhcp-agent trên Compute Node vừa cập nhật lên OVS-DPDK, sử dụng horizon hoặc command line để tạo lại.
  + 1. **Kiểm tra**
* Kiểm tra thư mục mount hugepage:

|  |
| --- |
| mount | grep Huge |

* Kiểm tra số lượng hugepages:

|  |
| --- |
| grep Huge /proc/meminfo |

* Kiểm tra memory:

|  |
| --- |
| free -m |

* Kiểm tra Open vSwitch:

|  |
| --- |
| ovs-vsctl show |

* Kiểm tra OVS-DPDK:

|  |
| --- |
| export RTE\_SDK=/usr/ /usr/src/DPDK-v17.05  export RTE\_TARGET= x86\_64-native-linuxapp-gcc  dpdk-devbind -s |

* + 1. **Dựng lại các máy ảo**
* Tạo các flavor enable hugepage, ví dụ tạo flavor với hugepage size là 2MB:

|  |
| --- |
| # create flavor  openstack flavor create --ram 4096 --disk 50 --vcpus 4 4C.4R.50D  openstack flavor create --ram 2048 --disk 15 --vcpus 1 1C.2R.15D  openstack flavor create --ram 2560 --disk 15 --vcpus 2 2C.2.5R.15D  # set hugepages  openstack flavor set 4C.4R.50D --property hw:mem\_page\_size=2MB  openstack flavor set 1C.2R.15D --property hw:mem\_page\_size=2MB  openstack flavor set 2C.2.5R.15D --property hw:mem\_page\_size=2MB |

Ngoài ra, để cho phép gán dedicated CPUs cho các VMs, thiết lập thêm các flavor key như sau:

|  |
| --- |
| openstack flavor set 4C.4R.50D --property hw:cpu\_policy="dedicated"  openstack flavor set 4C.4R.50D --property aggregate\_instance\_extra\_specs:pinned="true"  openstack flavor set 1C.2R.15D --property hw:cpu\_policy="dedicated"  openstack flavor set 1C.2R.15D --property aggregate\_instance\_extra\_specs:pinned="true"  openstack flavor set 2C.2.5R.15D --property hw:cpu\_policy="dedicated"  openstack flavor set 2C.2.5R.15D --property aggregate\_instance\_extra\_specs:pinned="true" |

* (Thao tác tùy chọn) Muốn launch các VM sử dụng chế độ multiqueues, bổ sung properties cho image hoặc volume sử dụng để boot VM:

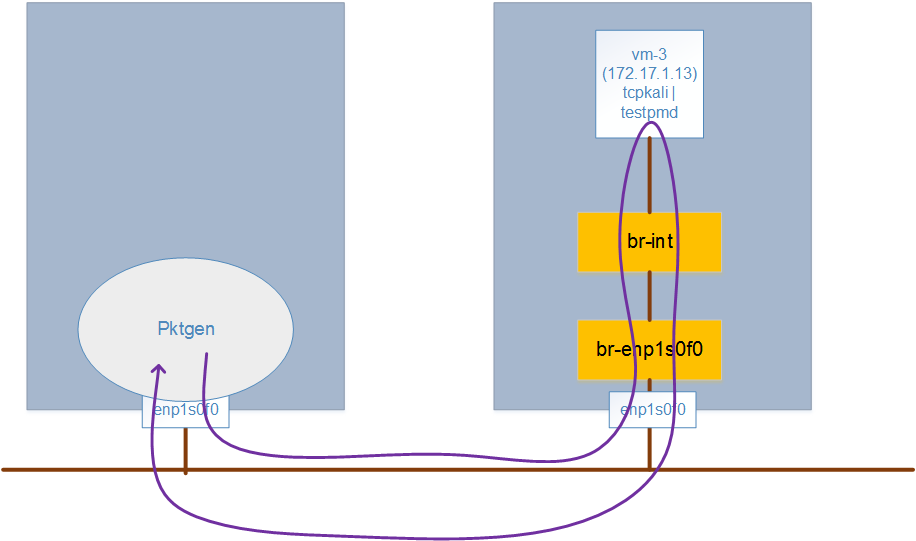
|  |
| --- |
| openstack image set IMAGE\_UUID --property hw\_vif\_multiqueue\_enabled="true"  openstack volume set VOLUME\_UUID --property hw\_vif\_multiqueue\_enabled="true" |

* Muốn launch các VM sử dụng OVS-DPDK phải lựa chọn các flavor enable hugepage.
* Với các VM cũ, để sử dụng OVS-DPDK, phải thực hiện các thao tác sau::
* Snapshot lại VM (Volume Snapshot hoặc Image snapshot)
* Thiết lập thuộc tính multiqueues cho Volume Snapshot hoặc Image snapshot theo hướng dẫn phía trên.
* Detach toàn bộ các interfaces của VM
* Tạo lại các Neutron port với fixed IP là các IPs hiện tại của VMs
* Xóa VM cũ đi và launch VM mới sử dụng snapshot vừa mới tạo (chú ý sử dụng flavor có cùng cấu hình với VM cũ và enable property là hugepage)
* Attach các Neutron port với fixed IP tương ứng vào VM (thao tác này sẽ tạo vhostuser interface cho VM trên OVS-DPDK bridge)
* Khởi động lại máy ảo và kiểm tra chức năng

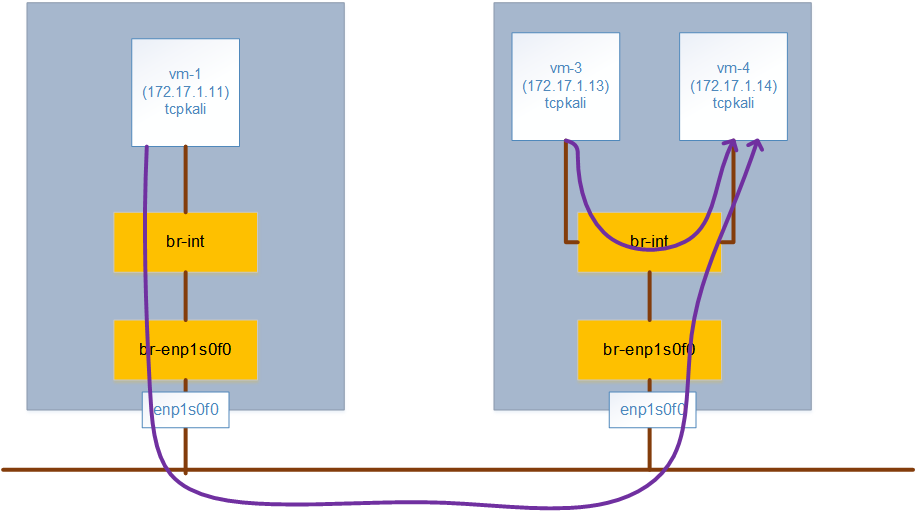
1. **So sánh hiệu năng của OVS-DPDK và OVS trên OpenStack**
   1. **Các testcases**

Các testcases sau đây áp dụng để đo tham số throughput (Mbps và Packets (frames/segments)/s)

* + 1. **North-South**

****

* + 1. **East-West**

****

Các bài test trong case E-W:

* **So sánh throughput của OVS và OVS-DPDK** trong hai trường hợp: hai VMs cùng host, hai VMs khác host
* **Kiểm chứng tính ổn định của OVS-DPDK** trong hai trường hợp: hai VMs cùng host, hai VMs khác host. Thời gian đo mỗi bài test: 12h.
  + 1. **Pipeline (WIP)**
  1. **Các công cụ đo và phương thức đo**
     1. **OpenStack với OVS DPDK**

|  |  |  |
| --- | --- | --- |
| Testcases | OVS | OVS-DPDK |
| L2 N-S | TCP kali | TCP kali, testpmd, pktgen, iperf |
| L2 E-W | TCP kali | TCP kali. iperf |
| L2 pipeline | N/A | ip\_pipeline |

* + 1. **Cấu hình các VMs và công cụ**
* Cấu hình các VM

|  |  |  |  |
| --- | --- | --- | --- |
| Testcases | Source | Destination | Instance flavor |
| L2 N-S | physical server E3-01 | vm-3 (trên physical server E3-02) | 3C.2.5R.15D |
| L2 E-W | vm-3 (E3-02) | vm-4 (E3-02) |
| L2 E-W | vm-1 (E3-01) | vm-4 (E3-02) |
| L2 pipeline | N/A | N/A |

* Cấu hình system và Open vSwitch (giống nhau trên cả hai server vật lý)

|  |  |
| --- | --- |
| Configuration | Value |
| NUMA nodes | 1 |
| Physical CPUs | 1 socket x 4 cores x 2 hyperthreads/cores = 8 CPUs |
| Hugepage size | 2MB |
| Number of hugepages | 3072 |
| OVS lcores (hyperthread) | 0 |
| OVS PMD CPUs (hyperthread) | 1,2,5,6 |
| OVS Socket Memory | 1024,0 (MB) |
| OpenStack instance CPUs (hyperthread) | 3,4,7 |
| Isolated CPUs | 1,2,3,4,5,6,7 |

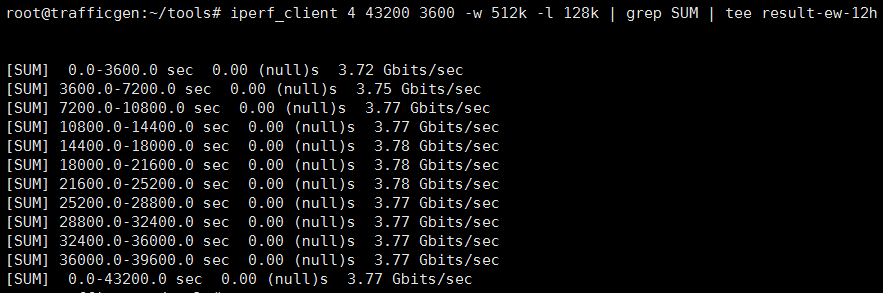
* + 1. **Commands**
  1. **Kết quả**
     1. **North-South**
* Số lần đo: 10
* Khoảng thời gian mỗi lần đo: 60s
* Direction: Unidirectional
* Kết quả:

|  |  |  |  |
| --- | --- | --- | --- |
| Lần đo | OVS (tcpkali) | OVS-DPDK (tcpkali)  Throughput (Bandwidth: Mbps/Packet rate: TCP MSS/op) | OVS-DPDK (pktgen-testpmd) |
| 1 |  | 5129.992 / 440277.9 |  |
| 2 |  | 5090.244 / 436866.6 |  |
| 3 |  | 5093.076 / 437109.7 |  |
| 4 |  | 5065.649 / 434755.7 |  |
| 5 |  | 5075.124 / 435568.9 |  |
| 6 |  | 5217.172 / 447760.1 |  |
| 7 |  | 5124.048 / 439767.7 |  |
| 8 |  | 5172.574 / 443932.5 |  |
| 9 |  | 5104.031 / 438049.8 |  |
| 10 |  | 5091.680 / 436989.8 |  |

* + 1. **East-West**
       1. Hai VM cùng node vật lý
* So sánh throughput OVS và OVS-DPDK
* Số lần đo: 10
* Khoảng thời gian mỗi lần đo: 60s
* Direction: Unidirectional
* Kết quả (Aggregate bandwidth / Packet (frame/segment) rate)

|  |  |  |  |
| --- | --- | --- | --- |
| Lần đo | OVS (tcpkali) | OVS-DPDK (tcpkali) | OVS-DPDK (pktgen-testpmd) |
| 1 | 1170.705 / 100514.4 | 4017.515 / 344802.0 |  |
| 2 | 1187.768 / 101998.9 | 4306.977 / 369644.1 |  |
| 3 | 1152.763 / 98979.1 | 3859.828 / 331267.9 |  |
| 4 | 1142.661 / 98156.8 | 4529.063 / 388703.7 |  |
| 5 | 1154.208 / 99147.8 | 3951.472 / 339133.2 |  |
| 6 | 1148.119 / 98601.3 | 4270.252 / 366492.2 |  |
| 7 | 1185.843 / 101885.6 | 4395.597 / 377249.8 |  |
| 8 | 1150.168 / 98754.8 | 4183.665 / 359061.7 |  |
| 9 | 1156.816 / 99328.5 | 4047.628 / 347386.4 |  |
| 10 | 1163.342 / 99912.4 | 4554.678 / 390905.8 |  |

* Kết quả đo tính ổn định của OVS-DPDK (đo througput giữa hai VMs cùng host trong 12h):
* Cấu hình mỗi VM: **Flavor 1C.2.R.15D** (1 CPUs, 2 GB RAM, 15 GB disk). Các extra properties: dedicated CPUs, multiqueues (cấu hình theo hướng dẫn ở mục 1.2.5)
* Cấu hình OVS-DPDK:
* Thời gian lấy mẫu: đo trong vòng 12h (43200s), cứ 1h (3600s) tính trung bình một lần.
* Kết quả: Throughput trung bình sau 12h là **3.77 Gbps**. Throughput trung bình mỗi giờ và trung bình sau 12h như sau:



* + - 1. Hai VM khác node vật lý

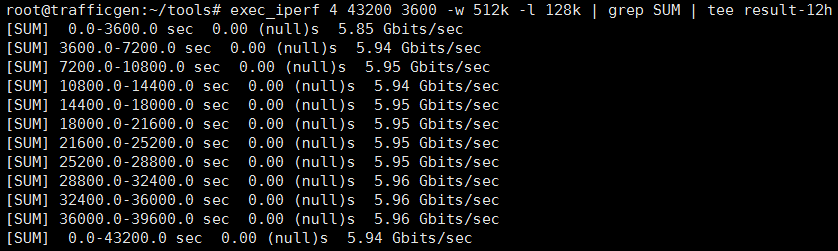
* So sánh throughput OVS và OVS-DPDK
* Số lần đo: 10
* Khoảng thời gian mỗi lần đo: 60s
* Direction: Unidirectional
* Kết quả:

|  |  |  |  |
| --- | --- | --- | --- |
| Lần đo | OVS (tcpkali) | OVS-DPDK (tcpkali) | OVS-DPDK (pktgen-testpmd) |
| 1 | 1026.665 / 88131.6 | 4111.942 / 352909.9 |  |
| 2 | 1045.616 / 89759 | 3830.745 / 328771.9 |  |
| 3 | 1025.456 / 88029.1 | 3992.677 / 342668.8 |  |
| 4 | 1010.660 / 86779.0 | 3669.774 / 314957.5 |  |
| 5 | 1001.476 / 86022.1 | 3794.867 / 325701.7 |  |
| 6 | 1004.714 / 86284.1 | 3722.620 / 319495.9 |  |
| 7 | 1059.749 / 91001.5 | 3829.529 / 328668.2 |  |
| 8 | 1057.925 / 90836.3 | 3657.449 / 313898.9 |  |
| 9 | 961.243 / 82556.4 | 4136.779 / 355037.0 |  |
| 10 | 1031.424 / 88560.4 | 4116.328 / 353287.8 |  |

* Kết quả đo tính ổn định của OVS-DPDK (đo througput giữa hai VMs khác host trong 12h):
* Cấu hình mỗi VM: **Flavor 2C.2.5.R.15D** (2 CPUs, 2.5 GB RAM, 15 GB disk). Các extra properties: dedicated CPUs, multiqueues (cấu hình theo hướng dẫn ở mục 1.2.5)
* Cấu hình OVS-DPDK: Cấu hình multiqueues cho các interfaces vật lý đã được attach vào OVS-DPDK và gán PMD cho các interfaces này:

|  |
| --- |
| # add dpdk port (attach physical interface)  ovs-vsctl add-port br-ex dpdk0 -- set Interface dpdk0 \  type=dpdk options:dpdk-devargs=0000:01:00.0  # set multiqueues && PMD  ovs-vsctl set interface dpdk0 options:n\_rxq=2 \  other\_config:pmd-rxq-affinity="0:2,1:3" |

* Thời gian lấy mẫu: đo trong vòng 12h (43200s), cứ 1h (3600s) tính trung bình một lần.
* Kết quả: Throughput trung bình sau 12h là **5.94Gbs**. Throughput trung bình mỗi giờ và trung bình sau 12h như sau:



* + 1. **Pipeline (WIP)**
  1. **Optimize OVS-DPDK Performance (WIP)**
     1. **Optimize OVS-DPDK**
* Isolate the CPU Cores
* Enable Huge Pages
* Enable TCP Segmentation Offload cho vhost-user port trên OVS-DPDK
* Enable tính năng Multiqueue Feature cho vHost-user và Physical DPDK Interfaces
* Enable Core Pinning và NUMA Awareness trên OpenStack Compute
  + 1. **Optiomize Guest**
* Thiết lập các properties 'extra\_specs' cho các VMs của OpenStack
* Enable Multiqueue cho VirtIO Interfaces

1. **Tham khảo**

* **[RedHats OVS-DPDK planning]:**

<https://access.redhat.com/documentation/en-us/red_hat_openstack_platform/11/html/network_functions_virtualization_planning_and_prerequisites_guide/assembly_ovsdpdk_parameters#c_ovsdpdk-cpu-parameters>

* **MTU:**

<https://docs.openstack.org/newton/networking-guide/config-mtu.html>

<https://www.openstack.org/assets/presentation-media/the-notorious-mtu.pdf>

* **Optimize OVS-DPDK:**

<https://builders.intel.com/docs/networkbuilders/optimizing_NFV_infrastructure_for_TCP_workloads.pdf>

1. **Phụ lục**

* **Hướng dẫn cài đặt ovs-dpdk:**

https://gitlab.visc.com/thaiph5/openvswitch/tree/ovs-dpdk

* **Hướng dẫn cài đặt và sử dụng các tools benchmark:**

<https://gitlab.visc.com/thaiph5/openvswitch/tree/ovs-dpdk/tools>