课程大作业 4

方天宬　519021910173

作业要求：

Hugepage in virtualization:

1. Prepare 2MB or 1GB hugepages on your host server. Present your hugepage configure
2. Create a QEMU KVM virtual machine using hugepages on the host
3. Create another QEMU KVM VM without hugepages
4. In both VMs allocate and use hugepages or not
5. Run memory instensive benchmark on two VMs and record the performance
6. Compare the result and try to give some explanation

一、打开 host 端 hugepage 功能

1. 开机时修改巨页

```
$ sudo vim /etc/libvirt/qemu.conf
$ # 取消注释 hugetlbfs_mount = "/dev/hugepages"
$
$ # 莫忘备份原来的 grub
$ sudo cp /etc/default/grub /etc/default/grub.copy
$
$ sudo vim /etc/default/grub
$  #  修 改  GRUB_CMDLINE_LINUX="transparent_hugepage=never
default_hugepagesz=1G hugepagesz=1G hugepages=5"
$ # 开了 5 个 1G 的大页面
$ # 更新 grub 并重启
$ sudo update-grub
$ sudo reboot
```

在修改完成后，输入 `cat /proc/meminfo | grep -i hugepage` 可查看 hugepage 的配置信息（见后文图）。

2. 用 dpdk 提供的 tools 进行修改

```
(base) ftc@FTC-COMPUTER:~$ cd 下载/dpdk-21.08/
(base) ftc@FTC-COMPUTER:~/下载/dpdk-21.08$ ls
ABI_VERSION  config    examples  MAINTAINERS        README
app          devtools  kernel    Makefile           usertools
build        doc       lib       meson.build        VERSION
buildtools   drivers   license   meson_options.txt
(base) ftc@FTC-COMPUTER:~/下载/dpdk-21.08$ cd usertools/
(base) ftc@FTC-COMPUTER:~/下载/dpdk-21.08/usertools$ ls
cpu_layout.py    dpdk-hugepages.py  dpdk-telemetry-client.py  meson.build
dpdk-devbind.py  dpdk-pmdinfo.py    dpdk-telemetry.py
(base) ftc@FTC-COMPUTER:~/下载/dpdk-21.08/usertools$
```

如图所示是 dpdk 的 usertool 文件夹，其中 dpdk-hugepages.py 是修改/查看巨页的脚本。

```
$ # 查看方法
$ python dpdk-hugepages.py -s
$ while true; do grep -i hugepage /proc/meminfo; sleep 1; done
$ # 修改方法，添加 5 张 1G 的巨页、256 张 2M 的巨页
$ sudo python3 dpdk-hugepages.py -p 1G -r 5G
$ sudo python3 dpdk-hugepages.py -p 2M -r 512M
```

```
(base) ftc@FTC-COMPUTER:~/下载/dpdk-21.08/usertools$ python dpdk-hugepages.py -sNode Pages Size Total
0    256   2Mb    512Mb
0    5     1Gb    5Gb

Hugepages mounted on /dev/hugepage /dev/hugepages
(base) ftc@FTC-COMPUTER:~/下载/dpdk-21.08/usertools$ grep -i hugepage /proc/meminfo
AnonHugePages:         0 kB
ShmemHugePages:        0 kB
FileHugePages:         0 kB
HugePages_Total:       5
HugePages_Free:        5
HugePages_Rsvd:        0
HugePages_Surp:        0
Hugepagesize:    1048576 kB
```

二、 启动虚拟机

使用命令行启动虚拟机时，务必使用 xml 开启 hugepage，否则会产生错误的现象（见附录）。本文给出三种启动虚拟机的方法：

1. 从 virt manager 启动

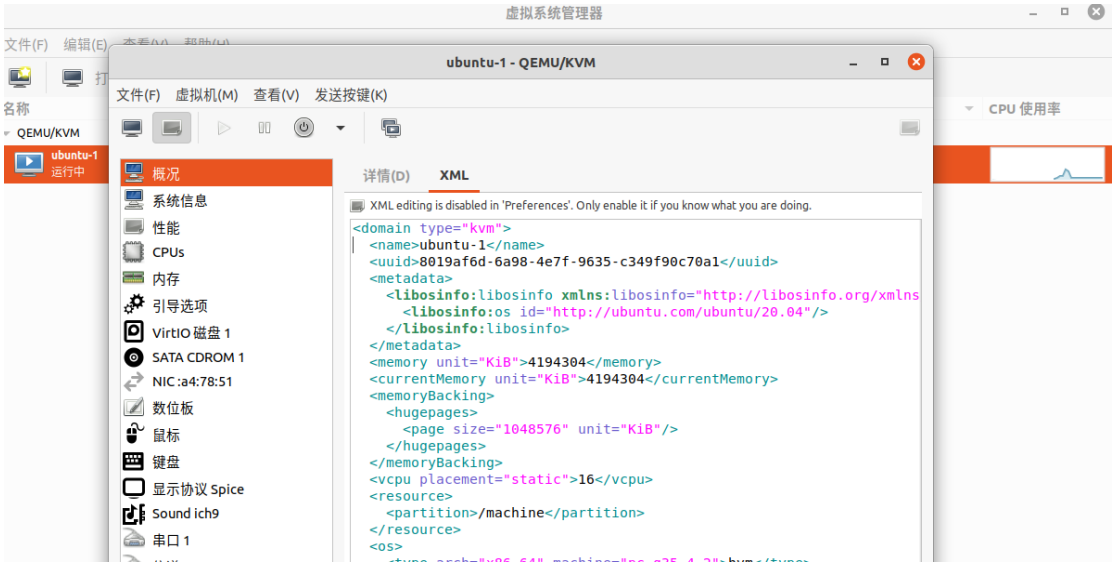
   输入命令 `sudo apt install virt-manager` 安装 `virt-manager`。打开其图形界面，按照指引操作：

2. 使用 `virsh` 命令启动

   在 `virt manager` 配置好信息后，打开其 xml 文件，在本地新建 xml 并复制到其中。输入以下命令创建并运行虚拟机：

```
$ # 1.创建
$ sudo virsh create ./ubuntu-1.xml
$ # 2.运行
$ virsh start --domain ubuntu-1
$ # 3.销毁
$ virsh undefine --domain ubuntu-1
```

xml 文件示例如图：



3. 编写 C 语言脚本启动

仍然在 virt manager 配置好虚拟机，打开其 xml 文件，在本地新建 xml 并复制到其中。创建 create.c 脚本，编译运行，源代码和编译参数如下：

```
    #include <libvirt.h>
    #include <stdio.h>
    #include <stdlib.h>

    char* fileToChar(const char* fileName){
        FILE* fp = fopen(fileName, "r");
        if(!fp){
            fprintf(stderr, "Error in open file %s\n", fileName);
            return NULL;
        }
        fseek(fp, 0, SEEK_END);
        long fileSize = ftell(fp);
        char* words = (char*) malloc(fileSize+1);
        fseek(fp, 0, SEEK_SET);
        fread(words, fileSize, sizeof(char), fp);
        fclose(fp);
        return words;
    }
    virDomainPtr defineDomainFromXML(virConnectPtr conn, const char* xmlFileName){
        char* xml = fileToChar(xmlFileName);
        if(!xml){ return NULL; }
        //        refer        to        https://libvirt.org/html/libvirt-libvirt-
domain.html#virDomainCreateFlags
        const int flags[] ={
                VIR_DOMAIN_NONE,
                VIR_DOMAIN_START_PAUSED,
                VIR_DOMAIN_START_AUTODESTROY,
                VIR_DOMAIN_START_BYPASS_CACHE,
                VIR_DOMAIN_START_FORCE_BOOT,
                VIR_DOMAIN_START_VALIDATE
        };
        virDomainPtr vm = virDomainCreateXML(conn, xml, flags[2]);
        free(xml);
        if(!vm){
            fprintf(stderr, "Error in create machine\n");
            return NULL;
        }
        return vm;
    }
    int main(){
        virConnectPtr conn = virConnectOpen("qemu:///system");
        const char* XML_FILE = "ubuntu-1.xml";
        virDomainPtr vm = defineDomainFromXML(conn, XML_FILE);
        char c;
        while (1){
```

```
            scanf("%c", &c);
            if(c=='c') break;
        }
        virDomainFree(vm);
        return 0;
    }
```

编译文件 CMakeLists.txt 为：

```
cmake_minimum_required(VERSION 3.5)
project(create_kvm)
add_executable(create_kvm
    create.c
)
target_include_directories(create_kvm
PUBLIC
    /usr/include/libvirt
)
link_directories(create_kvm
        /usr/lib/x86_64-linux-gnu
)
target_link_libraries(create_kvm
        # in /usr/lib/x86_64-linux-gnu
        libvirt-lxc.so
        libvirt-qemu.so
        libvirt.so
        libvirt-admin.so
)
add_compile_options(-Wall)
```

4. 命令行连接虚拟机

```
$ # 首先打开图形界面，使用图形界面里的命令行
$ virt-viewer -c qemu:///system ${your vm name}
$ # 打开图形界面的命令行，输入：
$ sudo systemctl enable serial-getty@ttyS0.service
$ sudo systemctl start serial-getty@ttyS0.service
$ # 关闭图形界面，在物理机打开命令行，尝试连接虚拟机
$ virsh console ${your vm name}
```

```
(base) ftc@FTC-COMPUTER:~$ virsh console ubuntu-1
Connected to domain ubuntu-1
Escape character is ^]

ftc@FTC-VM:~$ echo hello
hello
ftc@FTC-VM:~$
```

在启动虚拟机以后，巨页使用情况如下：

```
(base) ftc@FTC-COMPUTER:~/下载/dpdk-21.08/usertools$ grep -i hugepage /proc/meminfo
AnonHugePages:         0 kB
ShmemHugePages:        0 kB
FileHugePages:         0 kB
HugePages_Total:       5
HugePages_Free:        1
HugePages_Rsvd:        0
HugePages_Surp:        0
Hugepagesize:    1048576 kB
```

三、 进行 sysbench 测试

输入命令 `sudo apt install sysbench` 安装测试平台。统一使用如下命令测试:

```
$ sysbench --test=memory --threads=${threads} --memory-block-size=${mem_blksz} --memory-total-size=100G --memory-access-mode=${mode} run
```

1. 初步测试

HUGEPAGE	${threads}	${mem_blksz}	${mode}	speed (MB/s)
NO	1	8K	seq	10515.88
NO	2	8K	seq	12318.28
NO	4	8K	seq	13021.32
NO	1	1024K	seq	22140.77
NO	2	1024K	seq	42825.36
NO	4	1024K	seq	75688.65
YES	1	8K	seq	10811.94
YES	2	8K	seq	12714.87
YES	4	8K	seq	12867.31
YES	1	1024K	seq	21526.00
YES	2	1024K	seq	43022.91
YES	4	1024K	seq	71721.76
NO	1	8K	rnd	3203.03
NO	2	8K	rnd	2370.14
NO	4	8K	rnd	1675.95
NO	1	1024K	rnd	3241.47
NO	2	1024K	rnd	5450.97
NO	4	1024K	rnd	9241.85
YES	1	8K	rnd	3232.02
YES	2	8K	rnd	2407.42
YES	4	8K	rnd	1689.39
YES	1	1024K	rnd	3199.03
YES	2	1024K	rnd	5480.59
YES	4	1024K	rnd	9344.28

在做以上实验的时候, 我发现控制变量一定时, 多测几组, 每次结果相差得还是很大的。因此，只测一次得到的结果说服力不强。为了减小测试的 A 类不确定度，我对 threads=1 block_size=8K mode=seq 的测试选项进行了 20 次对比测试，结果如下表所示:

WITHOUT HUGEPAGE	18020.31	18048.11	18259.32	18392.91	18377.23
	18421.46	18468.20	18509.79	18370.58	18284.58
	18412.13	18316.36	18308.60	18075.28	17968.27
	18344.67	18457.85	18493.31	18459.09	18433.77
WITH HUGEPAGE	18435.46	18238.92	18318.17	18479.54	18276.78
	18464.31	18523.87	18576.67	18524.22	18501.47
	18505.61	18525.55	18499.69	18493.38	18351.77
	18444.58	18383.33	18489.49	18470.62	18439.98

在做这个实验的时候，我发现一个更离谱的现象：在测这 20 组数据的时候，我的鼠标是静止的，所以测得的结果比较整齐。当我测第 21 组时，我将鼠标从终端移动到虚拟机桌面点了一下，然后又到物理机外面点了一下，数据立马降到了 17841.74（如果此时打开浏览器，会降到 12339.67），降幅大概是 400MB/s，而且能够**稳定复现**。这个现象说明机器非常敏感，实验的设计有问题：**不应同时开两个虚拟机跑**，因为同时跑的话必然引起鼠标切换，导致误差，而这个误差会大于配置 hugepage 导致的差别。而且同时测试不能保证两个机器同时开始、同时结束，这会引起更大的误差。综上我认为：**应该保持每次测试的环境变化不太剧烈，分别配置测试，并且应该开启命令行测试而非图形界面。**

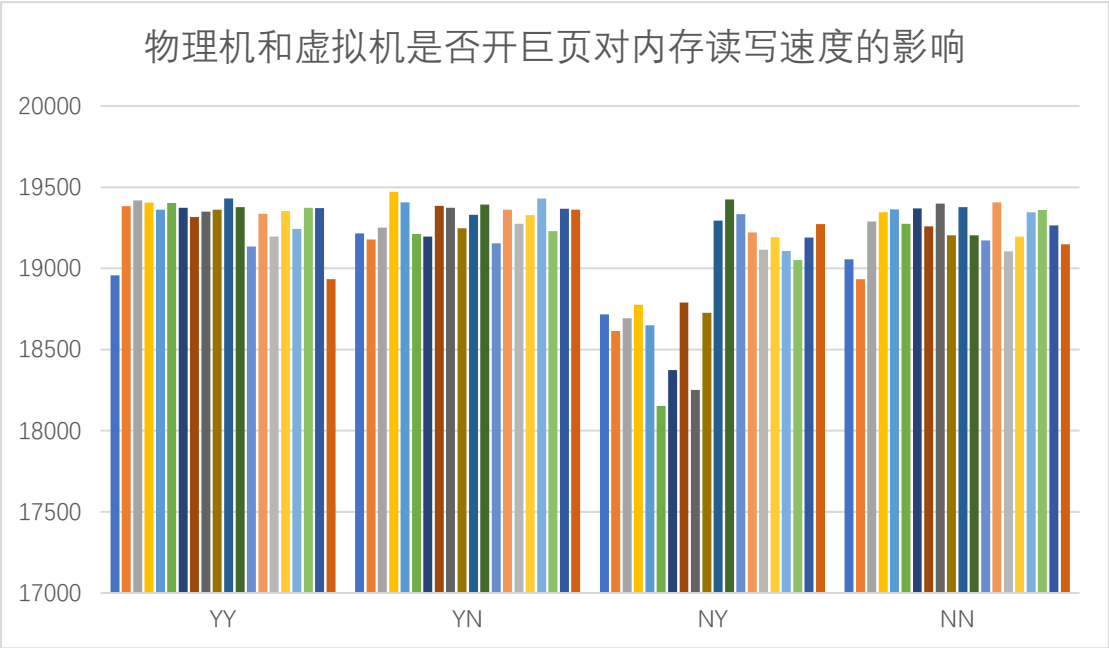
2. 深度测试

在进行深度测试的时候，本人只开启命令行连接，控制的变量为：本机是否开启 hugepage；虚拟机是否开启 hugepage；block_size。其他变量：${threads}=1，顺序读写，总读写容量 100G，block_size=8K,巨页大小 1G，开满内存，每组测试 20 次，开机后不开任何其他进程。

物理机 Y	虚拟机 Y	18957.07	19383.74	19419.41	19404.79	19361.74
		19403.16	19372.59	19315.39	19350.40	19362.06
		19430.57	19377.63	19135.43	19335.82	19195.25
		19354.12	19243.76	19373.26	19370.40	18933.45
	虚拟机 N	19214.97	19178.78	19250.16	19471.96	19407.20
		19211.58	19195.82	19384.13	19373.48	19247.79
		19330.64	19393.36	19154.67	19360.66	19274.26
		19328.08	19430.83	19229.97	19367.40	19361.85
物理机 N	虚拟机 Y	18715.90	18613.94	18693.23	18776.64	18649.69
		18152.71	18373.77	18789.24	18251.06	18727.02
		19294.07	19424.67	19334.23	19222.23	19114.77
		19192.26	19107.15	19050.85	19189.91	19272.59
	虚拟机 N	19056.47	18933.59	19288.50	19345.78	19363.14
		19274.13	19370.24	19258.81	19399.86	19204.46
		19377.14	19203.58	19172.30	19406.68	19105.10
		19196.07	19346.61	19359.06	19265.49	19148.93

四、 测试结果分析

1. 数据可视化



2. 得出结论

开不开 hugepage 在 5%的显著性水平下相差不大。

3. 分析结论

// TODO

五、 附录

1. 本机配置

CPU	Intel i7-10700K
显卡	NVIDIA RTX 3090
内存	Tigo 8GB DDR4 3200MHz x2 Gloway 8GB DDR4 3600MHz x2
固态硬盘	SAMSUNG MVZLB512HBJQ 500GB KIOXIA EXCERIA NVMe RC10 1000GB
机械硬盘	WDC WD2005VBYZ SATA 2TB WDC WD20EZAZ SATA 2TB
主板及芯片组	Intel Management Interface Intel SMBus-06A3
声卡	Realtek Audio
以太网卡	Realtek PCIe Family Controller
无线网卡	Intel Wi-Fi 6 AX201 160MHz
蓝牙	Intel Bluetooth

物理机操作系统： Ubuntu 20.04 x64，由 KIOXIA RC10 引导。

由于本机 CPU 的三缓较大（512KB、2MB、16MB），而且是原生 Linux（未套在虚拟环境运行），所以在做 sysbench 测试的时候速度飞快，特此说明。

2. 一种错误的启动方式

```
sudo qemu-system-x86_64 -m 4096 -smp 4 --enable-kvm -drive file=ubuntu20.qcow2 -vnc :5 -mem-path /dev/hugepages
```

上面的命令行参数提供了一种错误的启动方法，也是笔者原先犯的错误之一。这种方法指定了 qemu-system-x86_64 模拟器启动，没有使用 libvirt 所建议的方法（xml 启动）。因此这种方法对 libvirt 是不可见的，在笔者做前几次测试的时候失败了。

3. ubuntu-1.xml

```
    <domain type="kvm">
      <name>ubuntu-1</name>
      <uuid>8019af6d-6a98-4e7f-9635-c349f90c70a1</uuid>
      <metadata>
        <libosinfo:libosinfo
xmlns:libosinfo="http://libosinfo.org/xmlns/libvirt/domain/1.0">
          <libosinfo:os id="http://ubuntu.com/ubuntu/20.04"/>
        </libosinfo:libosinfo>
      </metadata>


      <memory unit="GiB">4</memory>
      <currentMemory unit="GiB">4</currentMemory>

    <!-- ====modify this==== -->
      <memoryBacking>
        <hugepages>
          <page size="1" unit="G"/>
        </hugepages>
      </memoryBacking>
    <!-- =============== -->

      <vcpu placement="static">16</vcpu>
      <os>
        <type arch="x86_64" machine="pc-q35-4.2">hvm</type>
        <boot dev="hd"/>
      </os>
      <features>
        <acpi/>
        <apic/>
        <vmport state="off"/>
      </features>
      <cpu mode="host-model" check="partial"/>
      <clock offset="utc">
        <timer name="rtc" tickpolicy="catchup"/>
        <timer name="pit" tickpolicy="delay"/>
        <timer name="hpet" present="no"/>
      </clock>
      <on_poweroff>destroy</on_poweroff>
      <on_reboot>restart</on_reboot>
      <on_crash>destroy</on_crash>
      <pm>
        <suspend-to-mem enabled="no"/>
        <suspend-to-disk enabled="no"/>
      </pm>
      <devices>
        <emulator>/usr/bin/qemu-system-x86_64</emulator>
        <disk type="file" device="disk">
          <driver name="qemu" type="qcow2"/>
          <source file="/media/ftc/H/ubuntu-1.qcow2"/>
          <target dev="vda" bus="virtio"/>
          <address     type="pci"     domain="0x0000"     bus="0x03"     slot="0x00"
function="0x0"/>
        </disk>
        <disk type="file" device="cdrom">
          <driver name="qemu" type="raw"/>
          <target dev="sda" bus="sata"/>
          <readonly/>
          <address type="drive" controller="0" bus="0" target="0" unit="0"/>
        </disk>
        <controller type="usb" index="0" model="ich9-ehci1">
          <address     type="pci"     domain="0x0000"     bus="0x00"     slot="0x1d"
function="0x7"/>
```

```
      </controller>
      <controller type="usb" index="0" model="ich9-uhci1">
        <master startport="0"/>
        <address type="pci" domain="0x0000" bus="0x00" slot="0x1d" function="0x0"
multifunction="on"/>
      </controller>
      <controller type="usb" index="0" model="ich9-uhci2">
        <master startport="2"/>
        <address    type="pci"    domain="0x0000"    bus="0x00"    slot="0x1d"
function="0x1"/>
      </controller>
      <controller type="usb" index="0" model="ich9-uhci3">
        <master startport="4"/>
        <address    type="pci"    domain="0x0000"    bus="0x00"    slot="0x1d"
function="0x2"/>
      </controller>
      <controller type="sata" index="0">
        <address    type="pci"    domain="0x0000"    bus="0x00"    slot="0x1f"
function="0x2"/>
      </controller>
      <controller type="pci" index="0" model="pcie-root"/>
      <controller type="pci" index="1" model="pcie-root-port">
        <model name="pcie-root-port"/>
        <target chassis="1" port="0x10"/>
        <address type="pci" domain="0x0000" bus="0x00" slot="0x02" function="0x0"
multifunction="on"/>
      </controller>
      <controller type="pci" index="2" model="pcie-root-port">
        <model name="pcie-root-port"/>
        <target chassis="2" port="0x11"/>
        <address    type="pci"    domain="0x0000"    bus="0x00"    slot="0x02"
function="0x1"/>
      </controller>
      <controller type="pci" index="3" model="pcie-root-port">
        <model name="pcie-root-port"/>
        <target chassis="3" port="0x12"/>
        <address    type="pci"    domain="0x0000"    bus="0x00"    slot="0x02"
function="0x2"/>
      </controller>
      <controller type="pci" index="4" model="pcie-root-port">
        <model name="pcie-root-port"/>
        <target chassis="4" port="0x13"/>
        <address    type="pci"    domain="0x0000"    bus="0x00"    slot="0x02"
function="0x3"/>
      </controller>
      <controller type="pci" index="5" model="pcie-root-port">
        <model name="pcie-root-port"/>
        <target chassis="5" port="0x14"/>
        <address    type="pci"    domain="0x0000"    bus="0x00"    slot="0x02"
function="0x4"/>
      </controller>
      <controller type="pci" index="6" model="pcie-root-port">
        <model name="pcie-root-port"/>
        <target chassis="6" port="0x15"/>
        <address    type="pci"    domain="0x0000"    bus="0x00"    slot="0x02"
function="0x5"/>
      </controller>
      <controller type="pci" index="7" model="pcie-root-port">
        <model name="pcie-root-port"/>
        <target chassis="7" port="0x16"/>
        <address    type="pci"    domain="0x0000"    bus="0x00"    slot="0x02"
function="0x6"/>
      </controller>
      <controller type="virtio-serial" index="0">
        <address    type="pci"    domain="0x0000"    bus="0x02"    slot="0x00"
function="0x0"/>
      </controller>
      <interface type="direct">
        <mac address="52:54:00:a4:78:51"/>
        <source dev="enp3s0" mode="bridge"/>
        <model type="e1000e"/>
        <address    type="pci"    domain="0x0000"    bus="0x01"    slot="0x00"
function="0x0"/>
      </interface>
      <serial type="pty">
        <target type="isa-serial" port="0">
```

```
          <model name="isa-serial"/>
        </target>
      </serial>
      <console type="pty">
        <target type="serial" port="0"/>
      </console>
      <channel type="unix">
        <target type="virtio" name="org.qemu.guest_agent.0"/>
        <address type="virtio-serial" controller="0" bus="0" port="1"/>
      </channel>
      <channel type="spicevmc">
        <target type="virtio" name="com.redhat.spice.0"/>
        <address type="virtio-serial" controller="0" bus="0" port="2"/>
      </channel>
      <input type="tablet" bus="usb">
        <address type="usb" bus="0" port="1"/>
      </input>
      <input type="mouse" bus="ps2"/>
      <input type="keyboard" bus="ps2"/>
      <graphics type="spice" autoport="yes">
        <listen type="address"/>
        <image compression="off"/>
      </graphics>
      <sound model="ich9">
        <address     type="pci"     domain="0x0000"     bus="0x00"     slot="0x1b"
function="0x0"/>
      </sound>
      <video>
        <model  type="qxl"  ram="65536"  vram="65536"  vgamem="16384"  heads="1"
primary="yes"/>
        <address     type="pci"     domain="0x0000"     bus="0x00"     slot="0x01"
function="0x0"/>
      </video>
      <redirdev bus="usb" type="spicevmc">
        <address type="usb" bus="0" port="2"/>
      </redirdev>
      <redirdev bus="usb" type="spicevmc">
        <address type="usb" bus="0" port="3"/>
      </redirdev>
      <memballoon model="virtio">
        <address     type="pci"     domain="0x0000"     bus="0x04"     slot="0x00"
function="0x0"/>
      </memballoon>
      <rng model="virtio">
        <backend model="random">/dev/urandom</backend>
        <address     type="pci"     domain="0x0000"     bus="0x05"     slot="0x00"
function="0x0"/>
      </rng>
    </devices>
  </domain>
```

六、 参考资料

1. dpdk 的脚本如何获取：https://core.dpdk.org/doc/quick-start/
2. xml 如何配置：https://libvirt.org/format.html
3. 关于 libvirt 的有关信息：https://wiki.libvirt.org/page/Main_Page