# **GVT-g Testenv setup on CentOS7.3**

Update Date:2018.01.18

## **Create or install CentOS7.3 in node**

Host we used is SKL Kontron server.

Use usb or DVD to install CentOS7.3 with GUI server enabled and Development tools installed.

## **Configure and setup host**

### Set proxy for host

1. #sed -i '$ a\\ export https\_proxy=https://proxy-prc.intel.com:911\n export http\_proxy=http://proxy-prc.intel.com:911' ~/.bashrc
2. #sed -i '$ a\\ export https\_proxy=https://proxy-prc.intel.com:911\n export http\_proxy=http://proxy-prc.intel.com:911' /home/media/.bashrc
3. #sed -i '$ a\\ proxy=https://proxy-prc.intel.com:911\n proxy=http://proxy-prc.intel.com:911' /etc/yum.conf
4. #reboot

### Upgrade kernel of Host server

1. **Copy PV3\_GVT package to your host user package.**

You can get it from [\\SHWDEJOINTD026\Media\_validation\GVT\PV3\_GVT](file:///\\SHWDEJOINTD026\Media_validation\GVT\PV3_GVT)

1. **Switch to root user and update Kernel**
   1. $su
   2. #cd PV3\_GVT/kernel/20180105
   3. [root@s2n6c2 kernel]# ls

kernel-4.11.0+-2.x86\_64.rpm

* 1. #rpm -Uvh kernel-4.11.0+-2.x86\_64.rpm --force --nodeps
  2. #reboot

1. **After host boot up, check the kernel if is 4.11.0+**
2. [media@s2n6c2 ~]$ uname -a

Linux s2n6c2 4.14.3+ #3 SMP Sat Jan 6 05:32:09 CST 2018 x86\_64 x86\_64 x86\_64 GNU/Linux

1. **Install needed Dependecies**
   1. # yum install -y git vim redhat-lsb SDL-devel.x86\_64 SDL.x86\_64 zip.x86\_64 zlib.x86\_64 zlib-devel.x86\_64 glib2-devel.x86\_64 pixman-devel.x86\_64 spice-server.x86\_64 spice-server-devel.x86\_64 spice-protocol.noarch uuid
2. **Edit Host Grub**
   1. #vim /boo t/gru2/grub.cfg

### END /etc/grub.d/01\_users ###

### BEGIN /etc/grub.d/10\_linux ###

menuentry 'CentOS Linux (4.14.3+) 7 (Core)' --class centos --class gnu-linux --class gnu --class os --unrestricted $menuentry\_id\_option 'gnulinux-3.10.0-514.el7.x86\_64-advanced-02876179-379a-4d62-955d-1ab524aef498' {

load\_video

set gfxpayload=keep

insmod gzio

insmod part\_msdos

insmod xfs

set root='hd0,msdos1'

if [ x$feature\_platform\_search\_hint = xy ]; then

search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos1 --hint-efi=hd0,msdos1 --hint-baremetal=ahci0,msdos1 --hint='hd0,msdos1' 08e23191-95cf-4439-8721-fcdb7552efc7

else

search --no-floppy --fs-uuid --set=root 08e23191-95cf-4439-8721-fcdb7552efc7

fi

linux16 /vmlinuz-4.14.3+ root=/dev/mapper/cl-root ro crashkernel=auto rd.lvm.lv=cl/root rd.lvm.lv=cl/swap rhgb quiet LANG=en\_US.UTF-8 console=ttyS0,115200n8 intel\_iommu=igfx\_off i915.enable\_gvt=1 drm.debug=0x2

initrd16 /initramfs-4.14.3+.img

}

* 1. #reboot

**NOTE:Marked yellow need to be added to grub file.**

1. **Build QEMU for KVMGT**

(as normal user)

* 1. $ git clone <https://github.com/01org/igvtg-qemu>
  2. $ cd igvtg-qemu/
  3. $ git checkout stable-2.9.0
  4. $ ./configure --prefix=/usr --enable-kvm --disable-xen --enable-debug-info --enable-debug --enable-sdl --enable-vhost-net --enable-spice --disable-debug-tcg --target-list=x86\_64-softmmu
  5. $ make -j8
  6. $sudo make install
  7. Copy bios.bin to /usr/bin
* $cd ~/PV3\_GVT/seabios
* $sudo cp bios.bin /usr/bin/
* $ sudo chmod 755 /usr/bin/ bios.bin
  1. Copy qemu-ifup and qemu-ifdown to your host /etc/

You can get from [\\SHWDEJOINTD026\Media\_validation\GVT](file:///\\SHWDEJOINTD026\Media_validation\GVT)

 

$sudo cp qemu-if\* /etc/

1. **Setup a proxy for external source download**

(as root user)

* 1. # touch /usr/bin/git-proxy
  2. # chmod 755 /usr/bin/git-proxy
  3. [root@s2n6c2 iHDV\_patch]# cat /usr/bin/git-proxy

[core]

proxy=proxy-shz.intel.com

exec socat STDIO SOCKS4:$proxy:$1:$2

* 1. # touch /home/media/.gitconfig
  2. [root@s2n6c2 media]# cat .gitconfig

[core]

gitproxy = none for intel.com

gitproxy = git-proxy

1. **Setup Bios**
   1. When restarting, press corresponding kerboard get into bios, then do following change.

chipset - system agent (SA) configuration - Graphics configuration -aperture size set from 256 to 1024 MB and GTT Size to 8MB

1. **Load GVT module**

(as normal user)

* 1. Add below command in /etc/dracut.conf under line“# additional kernel modules to the default”.

add\_drivers+="kvmgt vfio-iommu-type1 vfio-mdev"

* 1. $sudo mkinitrd /boot/initramfs-4.14.3+.img 4.14.3+ --force
  2. $sudo reboot
  3. After reboot, check the path **/sys/bus/pci/devices/0000\:00\:02.0/mdev\_supported\_types** if exist.

## **Install guest and configure guest**

### Create or install guest

1. Download Centos7.3 from mirrors.163.com or other source.

[**http://mirrors.163.com/centos/7.3.1611/isos/x86\_64/CentOS-7-x86\_64-DVD-1611.iso**](http://mirrors.163.com/centos/7.3.1611/isos/x86_64/CentOS-7-x86_64-DVD-1611.iso)

[**https://mirrors.aliyun.com/centos/7.3.1611/isos/x86\_64/CentOS-7-x86\_64-DVD-1611.iso**](https://mirrors.aliyun.com/centos/7.3.1611/isos/x86_64/CentOS-7-x86_64-DVD-1611.iso)

1. Use following command to install guest
2. #qemu-img create -f qcow2 test.qcow 50G
3. #/usr/bin/qemu-system-x86\_64 \

-m 2048 -smp 2 -M pc \

-name kvmgt -cpu host -hda /root/test.qcow \

-net nic -net tap,script=/etc/qemu-ifup \

-enable-kvm \

-machine kernel\_irqchip=on \

-net nic,model=e1000,macaddr=00:FE:EA:84:4F:dd \

-boot d \

-cdrom /home/media/gvt/CentOS-7-x86\_64-DVD-1611.iso

NOTE: Every GUEST should have no conflict macaddr

1. **Boot up guest and install mssdriver**
   1. **Build bridge on host**
      1. **Copy the crbr.sh to host**



You can also get it from

[\\SHWDEJOINTD026\Media\_validation\GVT](file:///\\SHWDEJOINTD026\Media_validation\GVT)

* + 1. **Build bridge**
       1. #sh crbr.sh sw0 add enp1s0f0
       2. #dhclient sw0

NOTE: The enp1s0f0 can be changed to your own machine network card. Everytime you reboot the host you need repeat the steps above.

* 1. **Boot up guest system**

# usr/bin/qemu-system-x86\_64 \

-m 2048 -smp 2 -M pc \

-name kvmgt -cpu host -hda /root/test.qcow \

-net nic -net tap,script=/etc/qemu-ifup \

-enable-kvm \

-machine kernel\_irqchip=on \

-net nic,model=e1000,macaddr=00:FE:EA:84:4F:dd \

* 1. **Copy PV3 related packages to guest**

You can get them from [\\SHWDEJOINTD026\Media\_validation\GVT\PV3\_GVT](file:///\\SHWDEJOINTD026\Media_validation\GVT\PV3_GVT)

* + 1. **Update guest kernel to 4.11.0 as root user**

$su

#cd PV3\_GVT/kernel/20180105

[root@localhost kernel]# ls

kernel-4.14.3+Display.x86\_64.rpm kernel-4.14.3+noDisplay.x86\_64.rpm

#rpm -Uvh kernel-4.14.3+Display.x86\_64.rpm --force –nodeps

#reboot

* + 1. **Update Guest Grub**

#cat /boot/grub2/grub.cfg

### END /etc/grub.d/01\_users ###

### BEGIN /etc/grub.d/10\_linux ###

menuentry 'CentOS Linux (4.14.3+) 7 (Core)' --class centos --class gnu-linux --class gnu --class os --unrestricted $menuentry\_id\_option 'gnulinux-3.10.0-514.el7.x86\_64-advanced-8c8a5d64-a6fa-4e16-8862-ffae4008198a' {

load\_video

set gfxpayload=keep

insmod gzio

insmod part\_msdos

insmod xfs

set root='hd0,msdos1'

if [ x$feature\_platform\_search\_hint = xy ]; then

search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos1 --hint-efi=hd0,msdos1 --hint-baremetal=ahci0,msdos1 --hint='hd0,msdos1' 12ebf343-ed89-478f-94aa-3641bff621f0

else

search --no-floppy --fs-uuid --set=root 12ebf343-ed89-478f-94aa-3641bff621f0

fi

linux16 /vmlinuz-4.14.3+ root=/dev/mapper/cl-root ro crashkernel=auto rd.lvm.lv=cl/root rd.lvm.lv=cl/swap rhgb LANG=en\_US.UTF-8 console=ttyS0,115200n8 ignore\_loglevel i915.enable\_hangcheck=0

initrd16 /initramfs-4.11.0+.img

}

NOTE:Marked yellow need to be added to grub file which enable serial port, guest log output,

* + 1. Install MSS driver and OPENCL

(as root user)

* + - 1. #cd /home/media/PV3\_GVT/MSS\_PV3
      2. #tar zxvf PV3.tar.gz
      3. #cd MediaServerStudioEssentials2017R3/SDK2017Production16.5.2/Generic
      4. #./install\_media.sh
      5. #tar xvf intel-opencl-cpu-r5.0-63503.x86\_64.tar.xz
      6. #tar xvf intel-opencl-r5.0-63503.x86\_64.tar.xz
      7. #tar xvf tar xvf intel-opencl-devel-r5.0-63503.x86\_64.tar.xz
      8. #cp -raf etc/\* /etc/
      9. #cp -raf opt/intel/opencl/ /opt/intel/
      10. #ldconfig
      11. Close qemu tool on host

NOTE:When execute step4, if the machine is SKL, choose y, other choose n, after the guest started normally again, the vainfo command may show unconnect info, try command “unset DISPLAY”

1. **Create VGPU on host and boot up guest**

(as root user)

* 1. #uuid

bb6b31ce-6cfb-11e7-adc4-00a0a597e7e0

* 1. Create vgpu uuid for specific use

1. Create uuid for only one guest in host

#echo “bb6b31ce-6cfb-11e7-adc4-00a0a597e7e0” > /sys/bus/pci/devices/0000\:00\:02.0/mdev\_supported\_types/i915-GVTg\_V5\_1/create

1. Create uuid for three guest

#echo “bb6b31ce-6cfb-11e7-adc4-00a0a597e7e1” > /sys/bus/pci/devices/0000\:00\:02.0/mdev\_supported\_types/i915-GVTg\_V5\_4/create

#echo “bb6b31ce-6cfb-11e7-adc4-00a0a597e7e2” > /sys/bus/pci/devices/0000\:00\:02.0/mdev\_supported\_types/i915-GVTg\_V5\_4/create

#echo “bb6b31ce-6cfb-11e7-adc4-00a0a597e7e3” > /sys/bus/pci/devices/0000\:00\:02.0/mdev\_supported\_types/i915-GVTg\_V5\_4/create

* 1. #ls /sys/bus/pci/devices/0000\:00\:02.0/

backlight consistent\_dma\_mask\_bits driver i2c-0 local\_cpulist msi\_bus reset revision vendor

bb6b31ce-6cfb-11e7-adc4-00a0a597e7e0 current\_link\_speed driver\_override i2c-1 local\_cpus msi\_irqs resource rom

boot\_vga current\_link\_width drm i2c-2 max\_link\_speed numa\_node resource0 subsystem

broken\_parity\_status d3cold\_allowed enable index max\_link\_width power resource2 subsystem\_device

class device firmware\_node irq mdev\_supported\_types remove resource2\_wc subsystem\_vendor

config dma\_mask\_bits gvt\_firmware label modalias rescan resource4 uevent

* 1. boot up guest

#/usr/bin/qemu-system-x86\_64 -m 16384 -smp 2 -M pc -name kvmgt3 -cpu host -hda /root/test.qcow -net nic -net tap,script=/etc/qemu-ifup -enable-kvm -machine kernel\_irqchip=on -serial file:/home/guest.log -net nic,model=e1000,macaddr=00:FE:EA:84:4F:dc -device vfio-pci,sysfsdev=/sys/bus/pci/devices/0000:00:02.0/ bb6b31ce-6cfb-11e7-adc4-00a0a597e7e0,rombar=0

**NOTE1:after creating new VGPU uuid, remember to update new uuid to the bootup script. The mem value can adjust from your host mem, it makes sure guest has enouth resource to use,**

**NOTE2:how many guests can support depend on the gpufrequency size of your machine.**

**NOTE2:if you want change mem size and cpus, just change the value of “-m” and “-smp” in script .**

1. **Replace the guest iHD\_drv\_video.so**
   1. #cd /home/media/ PV3\_GVT/iHDV\_patch/
   2. #cp iHD\_drv\_video.so /opt/intel/mediasdk/lib64/
2. **Install perl, third party tool on guest**

(as root user)

* 1. Copy autoinstall scripts to the guest

You can get from \\SHWDEJOINTD026\Media\_validation\CGSS\MSDK Test Enviornment install script\msdkenv\_autoinstall

* 1. #cd msdkenv\_autoinstall
  2. comment out from “#prepare test\_system” to the last line.
  3. #./install\_all.sh
  4. #reboot

NOTE:Because we have limited space on guest, we need mount the test system from host machine.After execute install\_all.sh, you need mannuly check if the tools are correctly installed from install\_version.log

### Run MSDK function or performance test

1. **Setup on host**

(as root user)

* 1. #yum install nfs-utils -y
  2. #vim /etc/exports

/home/media/test\_system \*(insecure,rw,async,no\_root\_squash)

* 1. # systemctl enable rpcbind.service
  2. #systemctl enable nfs-server.service
  3. #systemctl start rpcbind.service
  4. #systemctl start nfs-server.service
  5. #systemctl disable firewalld.service
  6. #systemctl restart iptables.service

(as normal user)

* 1. $cd /home/media/ PV3\_GVT/
  2. $ unzip test\_system.zip -d /home/media/test\_system
  3. $chmod 775 -R /home/media/test\_system
  4. Copy sample app to test\_system
     1. $cd ~/PV3\_GVT/MSS\_PV3/MediaServerStudioEssentials2017R3/MediaSamples\_Linux\_2017R3\_b698/samples/\_bin/x64
     2. $sudo cp \*.so ocl\_rotate.cl sample\_\* /home/media/test\_system/build/lin\_x64/bin
  5. Copy 6.0 tool to test\_system

You can get from [\\SHWDEJOINTD026\Media\_validation\validation\_tools\6.0](file:///\\SHWDEJOINTD026\Media_validation\validation_tools\6.0)

$sudo cp -r 6.0 /home/media/test\_system

**NOTE: Everytime you reboot your host, you need repeat step g and h, or you will not find the mountserver.**

1. **Setup on guest**

(as normal user)

* 1. $mkdir -p /home/media/ws/msdk\_validation/mediasdk\_streams
  2. $mkdir -p /home/media/ws/msdk\_validation/test\_system
  3. $chmod 775 -R /home/media/ws
  4. $showmont -e HOST\_IP

Export list for HOST\_IP:

/home/media/test\_system \*

(as root user)

* 1. #mount -t nfs -o nolock,nfsvers=3,vers=3 -o proto=tcp  HOST\_IP:/home/media/test\_system /home/media/ws/msdk\_validation/test\_system/
  2. #mount -t nfs -o vers=3 10.67.116.91:/datadisk/streams /home/media/ws/msdk\_validation/mediasdk\_streams

1. **Run msdk case on guest**
   1. Enter the test\_system package on guest

#cd /home/media/ws/msdk\_validation/test\_system

* 1. #export MEDIASDK\_STREAMS=/home/media/ws/msdk\_validation/mediasdk\_streams
  2. #export MEDIASDK\_ROOT=/home/media/ws/msdk\_validation/test\_system
  3. #export LD\_LIBRARY\_PATH=./:/user/local/lib:/usr/lib64:/opt/intel/mediasdk/lib64:/opt/intel/common/mdf/lib64:/opt/intel/opencl:/home/media/ws/msdk\_validation/test\_system/build/lin\_x64/bin
  4. #perl test\_driver.pl -s h264d\_common -p c7.3\_skl\_64\_server --copy\_missed\_streams

### Run umd performance lagency test

1. Disable CPU Turbo option in bios.
2. Get newest umd lagency performance test version from media-storage server

$pwd

[media@guest\_s2n6c2 ~]$ pwd

/home/media

$ git clone media@media-storage:/datadisk/repos/perf\_test

1. Create needed path

#mkdir -p /home/intel/content-performance

#mkdir -p /home/intel/work

#mkdir -p /opt/remote

#chmod 775 -R /home/intel

#ln -s /home/media/perf\_test/skl-performance /home/intel/work/skl-performance

1. Add ***intel\_pstate=disable*** to grub file reference to below.

#vim /boot/grub2/grub.conf

**### BEGIN /etc/grub.d/10\_linux ###**

menuentry 'CentOS Linux (4.14.3+) 7 (Core)' --class centos --class gnu-linux --class gnu --class os --unrestricted $menuentry\_id\_option 'gnulinux-3.10.0-514.el7.x86\_64-advanced-8c8a5d64-a6fa-4e16-8862-ffae4008198a' {

load\_video

set gfxpayload=keep

insmod gzio

insmod part\_msdos

insmod xfs

set root='hd0,msdos1'

if [ x$feature\_platform\_search\_hint = xy ]; then

search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos1 --hint-efi=hd0,msdos1 --hint-baremetal=ahci0,msdos1 --hint='hd0,msdos1' 12ebf343-ed89-478f-94aa-3641bff621f0

else

search --no-floppy --fs-uuid --set=root 12ebf343-ed89-478f-94aa-3641bff621f0

fi

**linux16 /vmlinuz-4.14.3+ root=/dev/mapper/cl-root ro crashkernel=auto rd.lvm.lv=cl/root rd.lvm.lv=cl/swap rhgb quiet LANG=en\_US.UTF-8 ignore\_loglevel i915.enable\_hangcheck=0 intel\_idle.max\_cstate=1 intel\_pstate=disable**

**initrd16 /initramfs-4.14.3+.img**

**}**

#reboot

1. Update mediasdk version in skl-performance package.
2. Check msdk version in guest

$ strings /opt/intel/mediasdk/lib64/libmfxhw64.so | grep version

mediasdk\_product\_version: 7.0.16053710

mediasdk\_file\_version: 7.17.6.13

mfxExtVPPFrameRateConversion

28mfxExtVPPFrameRateConversion

31MFXVideoVPPColorSpaceConversion

30MFXVideoVPPFrameRateConversion

32MFXVideoVPPVideoSignalConversion

\_\_itt\_api\_version

ColorSpaceConversion

N3UMC20ColorSpaceConversionE

.gnu.version

.gnu.version\_d

.gnu.version\_r

1. Get 7.0.16053710 mediasdk from remote server.

# mount -t nfs 10.239.141.10:/opt /opt/remote

# /opt/remote/MSDK

#rm -rf /home/intel/work/skl-performance/imports/

# cp Linux\_MSS2017R3-PV\_2017\_WW24.2\_b710/lin\_x64/mediasdk home/intel/work/skl-performance/imports/

(this step determine which package depends on msdk version last 3 number)

1. Get clips streams for UMD lagency performance

# mount -t nfs 10.67.113.77:/home/media/long/umd\_streams/content-performance content-performance/

1. Run the lagency performance case

#cd /home/intel/work/skl-performance

1. Run all cases

# ./ LucasRun\_16.3\_zhipeng\_all\_cpu\_new.sh

1. Run specific suite

# ./LucasRun\_16.3\_zhipeng\_all\_cpu\_new.sh CVS\_PPQ\_LiteSuite\_1N\_16.3\_4k.csv

1. Run only one case

#./lucas -s CVS\_PPQ\_LiteSuite\_1N\_16.3\_4k.csv CASE\_ID

Note: after test completed, it generate a Summary\*.csv file, include fps , frames, u + s , u, s cpuusage information during test.

### Run stress test

1. Get scripts from git server and copy it to your test machine.

$git clone ssh://<USERNAME>@git-ccr-1.devtools.intel.com:29418/npg\_media\_validation-npg\_media\_validation-validation\_msdk\_stress.git stress

You can also get from [\\SHWDEJOINTD026\Media\_validation\Stress](file:///\\SHWDEJOINTD026\Media_validation\Stress)

1. Mount the stress streams

# mount -t nfs -o vers=3 10.67.116.91:/datadisk/streams mediasdk\_streams

1. Change the count number in smt\_stress.sh

#vim smt\_stress.sh

#Start test

for ((i=1;i<=6350;i++ ))

1. Run the stress test

#pwd

/home/media/stress

#ls -l

total 3368

-rwxrwxr-x 1 media media 1085 Jan 5 15:19 check\_result.sh

-rwxrwxr-x 1 media media 2294 Jan 5 15:21 generate\_par.sh

drwxrwxr-x 2 media media 6 Jan 5 15:38 mediasdk\_streams

-rwxrwxr-x 1 media media 83 Jan 5 15:19 mount.sh

-rw-rw-r-- 1 media media 978 Jan 5 15:19 Readme.txt

-rwxrwxr-x 1 media media 437096 Jan 5 15:19 sample\_multi\_transcode

-rwxrwxr-x 1 media media 4239 Jan 11 11:22 smt\_stress.sh

-rwxrwxr-x 1 media media 2981370 Jan 5 15:19 sysinfo

-rwxrwxr-x 1 media media 1172 Jan 5 15:19 sysinfo.sh

#./smt\_stress.sh