GPU Accelerated VM with KVM

research is ongoing.

- https://medium.com/@calerogers/gpu-virtualization-with-kvm-qemu-63ca98a6a172
- http://www.virtualopensystems.com/en/services/vfio-full-gpu-virtualization/

motivation

a while a go i've moved from hand drawings to inkscape drawings to finally using CAD for planning my woodworking projects. I greatly enjoy the whole CAD design process prior to actually start cutting wood. however, there is one problem: all good CAD programs that are available for reasonable money and easy enough to operate for a CAD novice are Windows and OSX applications. There is FreeCAD on Linux which is awesome for a free tool but it just isn't quite there yet. Or maybe my skills aren't quite there yet :) .. I started with FreeCAD, then moved to the Windows side with SketchUP and now continued to Fusion 360 which i like alot. Autodesk is developing a fully web-based gui for Fusion 360 however this seems to take its time, so for now i am stuck with the Windows version of it.

Since the 3D support for vm's is inexistent with KVM/qemu and very poor and buggy with VirtualBox i am currently using a vmware player based Windows 10 VM which performs quite well. I can wor on my CAD 3d models comfortably fast but my host system gets extreamly loud and both the CPU and GPU get loaded heavily while using the VM. There are also stability issues. Booting the vm i usually have about a 50/50 chance of my host OS surviving. Every second time my host system freezes and i have to hard-reset to reboot it.

There is another big downside of all this: if I want to draw a little design for a part i need to 3d print for work, i have to go home and do it there or i have to use some crappy vnc connection to my home pc to work on my cad this way.. however, vmware player isn't really ment to be used as a server and this is clearly noticable when trying to do so :) .. so i would need to replicat the whole vm setup on every machine i want to use cad.

to solve these problems my goal is now to run a windows 10 vm on my ubuntu based server which i have in my cellar. I want to use KVM with a passthrough GPU so the vm has a real GPU to work with and can provide all the 3d rendering glory without limitations and rendering errors to the software. i then want to be able to access that VM from some kind of remote-desktop or even better remote-app client from all my linux computers in the office and at home or on the go.. it does not need to be available offline, i'm fine with being limited to places where i have a network connection to the server or a vpn through the internet.

first try: citrix XenDesktop

which was actually renamed while i was playing around with it.. it is now called virtual apps and desktop or something like that.. i don't see anything virtual about xenDesktop, as it does not do virtualisation by itself it just runs on physical or virtual machines.

here are my reasons why i am trying this one out first:

- i've been working with other xen desktops to get remote access to customer systems and i like the way it handles. especially because it has one of the key features i expect from a remote desktop solution: the size of the remote desktop should automatically adjust to the window size as i ressize the window.. and citrix does that very well
- they promise to support 3d rendering with pass-through gpu's on vm's, and that's exactly what i am planing on doing
- possibly the xenDesktop package also supports sharing just apps which would be even better as i am really only interested in having one windows application available on multiple computers without needing to install a new vm on every one of these computers. however, i don't understand why xenApp which supposedly is ONLY providing remote apps seems to be more expensive.. I guess i will eventually find out ;)
- i find xenDesktop is reasonably priced. by looking at the comparison between the different editions it seems that xenDesktop Enterprise edition is more than enough for me but VDI seems to little. Since I only need a 1-user license i can get away with 240$ for a perpetual license. with xen apps i would need to buy a concurrent user license which is more expensive (about 100$ more). NICE DCM for example would cost 300$ plus an annual 60$ support contract.

here are some catches i have found so far:

- a downside is, i also have to facture in the price of a windows server license, as xenDesktop needs to run on a windows server, it does not want to install all its management software pieces on a regular windows 10 installation. but it will run on the older 2012r2 which is available for a little over 100$ for the essentials edition in some online shops by now. this however, makes it more expensive than NICE DCM at least for the initial purchase.
- stupidly, xen studio, which seems to be needed to manage the whole xenDesktop stuff, needs to run on a server which is joined into a windows domain. however, and here comes the really stupid part, the xen Desktop stuff can't be installed on a windows server that is an AD Domain Controller! this essentially means, that besides the windows server for the xen stuff i need ANOTHER windows server just as a domain controller..
    - i will try to work around this problem by using samba ~~on my host server~~ in an LXC container running ubuntu on my host server to provide a windows domain. (i have samba shares on my host-server and the samba guys recommend not to have both a domain controller and samba shares on the same machine.. plus i don't really want to have a windows domain at home, so i don't want to manage my samba shares via a domain)

so here is the setup i am going to try out:

- my already existing physical ubuntu based server will host two KVM/qemu virtual machines and one LXC container:
    - one windows server 2012r2 for xenDesktop management
    - one windows 10 with GPU Passthrough
    - one ubuntu LXC Container running Samba as an Active Directory Domain Controller
- I will first test to get the whle xen stuff to work without the gpu installed, if that works, i'll try to get the gpu passthrough to work with it.
- at the very beginnig my windows 10 vm will be running as a vmware player vm on my client machine which again is natively running ubuntu. this is for the simple reason that i am currently using that exact vm to run my cad software (fusion 360) 3d acceleration, so i'll just use that and then port it to KVM and then add gpu passthrough

pastebin for some snippets from my current setup

booting the windows vm with two cd-roms to install windows 2012r2 and virtIO storage devices at the same time. i currently run this manually in a “screen” session

```
qemu-system-x86_64 -enable-kvm -k de-ch -name winsrv -drive
file=/bulk/vm/winsrv.img,if=virtio,format=raw,index=0 -m 4096 -device
virtio-net-pci,netdev=net0,mac=52:54:00:01:01:01 -netdev tap,id=net0 -cdrom
9600.16384.WINBLUE_RTM.130821-1623_X64FRE_SERVER_SOLUTION_EN-US-
IRM_SSSO_X64FRE_EN-US_DV5.ISO -drive file=virtio-win.iso,media=cdrom,index=1
-vnc :1 -monitor tcp:127.0.0.1:5802,server,nowait -boot once=d -smp
cpus=4,cores=4 -cpu host,kvm=off -monitor vc
```

(i currently have a rather underequipped host system as i haven't planned on running vm's on it when i built it.. so what you see here is really the minimum for these services.. anything below 4GB ram will end in heavy swapping.

samba setup (inside lxc container)

```
apt install lxc
echo "lxc.lxcpath = /bulk/vm" >> /etc/lxc/lxc.conf
lxc-create --template ubuntu --name=dc1
sed -i 's/lxcbr0/br0/' /bulk/vm/dc1/config
lxc-start -n dc1 -F
sudo su -
adduser <username>
adduser <username> sudo
exit
exit
#login as new user
sudo su -
userdel -r ubuntu
apt install man nano samba krb5-user winbind
```

`/etc/network/interfaces` should look something like that (with a static ip)

```
auto eth0
iface eth0 inet static
    address 192.168.168.7
    netmask 255.255.255.0
    network 192.168.168.0
    broadcast 192.168.168.255
    gateway 192.168.168.254
```

also fix your `resolv.conf` to point to a working dns and search in our new domain. to do so, don't edit `/etc/resolv.conf` but edit `/etc/dhcp/dhclient.conf` instead. un-comment and edit these two lines:

```
supersede domain-name "psdom.psuter.ch";
prepend domain-name-servers 192.168.168.7;
```

```
reboot
```

after changing the network config

make sure that the timezone is configured correctly, as your ad-dc server will also act as a time server for the machines in the domain. if `date` does not display the correct timezone, change it using

```
dpkg-reconfigure tzdata
```

in my case the root directory of the lxc container is inside a zfs volume. since samba ad service needs acl's i had to enable that on my zfs volume first:

```
zfs set acltype=posixacl storage/bulk
```

I'm follwing the Samba Wiki

my vm host system (and therefore samba AD controller) has the ip: `192.168.168.1`

```
systemctl stop samba-ad-dc.service
# the following line is commented out because it is dangerous! test it with
an echo first, to make sure
# DDD/* is not interpreted as /* as this WILL run an "rm -rf /*" which is
BAD
# smbd -b | egrep "LOCKDIR|STATEDIR|CACHEDIR|PRIVATE_DIR" | awk '{print $2}'
| xargs -I DDD /bin/bash -c 'rm -rf "DDD/*"'
sed -i 's/^.*dc1.*$/192.168.168.7   dc1.psdom.psuter.ch dc1/' /etc/hosts
rm /etc/samba/smb.conf
#i will be using bind as my dns, since i already have it installed and
configured anyway.
  samba-tool domain provision --use-rfc2307 --realm=psdom.psuter.ch --
domain=psdom --server-role=dc --dns-backend=SAMBA_INTERNAL --
adminpass=**REDACTED**
```

samba-tool will create the domain. **tipp** add a space at the begining of the command line so your dns password will not be saved cleartext in your bash history.. you can also use `–interactive` besides the `–use-rfc2307` parameter to enter all these details including the password interactively

```
cp /var/lib/samba/private/krb5.conf /etc/
systemctl restart samba-ad-dc
systemctl status samba-ad-dc
```

now test your setup:

```
smbclient -L localhost -U%
Domain=[PSDOM] OS=[Windows 6.1] Server=[Samba 4.3.11-Ubuntu]

    Sharename       Type      Comment
```

```
    ---------       ----      -------
    netlogon        Disk
    sysvol          Disk
    IPC$            IPC       IPC Service (Samba 4.3.11-Ubuntu)
Domain=[PSDOM] OS=[Windows 6.1] Server=[Samba 4.3.11-Ubuntu]

    Server               Comment
    ---------            -------

    Workgroup            Master
    ---------            -------
    WORKGROUP            PSUTER
```

if you get an error saying NT_STATUS_OBJECT_NAME_NOT_FOUND make sure you have winbind installed and reboot

```
smbclient //localhost/netlogon -UAdministrator -c 'ls'
Enter Administrator's password:
Domain=[PSDOM] OS=[Windows 6.1] Server=[Samba 4.3.11-Ubuntu]
  .                                   D        0  Sat Aug 25 23:50:10 2018
  ..                                  D        0  Sat Aug 25 23:50:15 2018

        11714059904 blocks of size 1024. 4625221248 blocks available

root@dc1:~# host -t SRV _ldap._tcp.psdom.psuter.ch
_ldap._tcp.psdom.psuter.ch has SRV record 0 100 389 dc1.psdom.psuter.ch.
root@dc1:~# host -t SRV _kerberos._udp.psdom.psuter.ch
_kerberos._udp.psdom.psuter.ch has SRV record 0 100 88 dc1.psdom.psuter.ch.
root@dc1:~# host -t A dc1.psdom.psuter.ch
dc1.psdom.psuter.ch has address 192.168.168.7
```

to test the dns stuff

and finally test authentication

```
kinit administrator
```

Citrix Installation

After installing Windows Server 2012r2 I had to first complete the server configuration wizard. make sure your windows server does not become a Domain Controller. If it does, demote it before you continue. Then Join the machine into our Samba Domain like you join any windows machine into a domain. In the network config, set the Samba AD-DC's IP as DNS. I had to install some rounds of updates before the installer on the iso image would let me start installing Citrix.

I then simply followed one step after the other through the intaller. This needed several reboots. After the reboot i had to point again to the installation sources which where on an ISO image which was no longer mounted after boot. I had to right clikc on a random folder and select “Open in new Window” so i would get a normal explorer window in order to mount (double click) the ISO image so i could

then select the mounted image in the original dialog window.

It would have been easier to extract the ISO image to a folder and install from there. Make this a shared folder, as you will also need to run the same installer to install the virtual desktop agent on the windows 10 machine.

after completing the installation wizard and about 3 reboots you should finally have a couple of Citrix applications in your start menu. If you haven't, run the installer again, it will continue from where it has left off.

once you have all these programs in your start menu, begin with the `Citrix Studio` and complete the first three steps it asksk you for in the main window. Then click on the `Configuration–>StoreFront` item and create a new StoreFront server. After that, continue on the `Citrix StoeFront` entry at the bottom of the list and set up your StoreFront.

Chose to make that Store your default Webpage. On my System that didn't work, Citrix StoreFront did not have access to the necessary config of IIS. It worked on my first Try on a Windows Server 2016 though.

after settign this up, you can then open https://your.server.fqdn in a web-browser. If StoreFront is not the defualt webpage, then the url will be https://your.server.fqdn/Citrix/StoreWeb/.

After logging in you should see your Windows 10 machine and you should be able to start a session. That is, if no one is loged in anymore to your windows 10 machine other than your own user.

Tweaking the linux client

in order to get the middle mous button (wheel) to click, i had to modify the config of my citrix receiver on my client machine. I had to edit the file `$HOME/.ICAClient/wfclient.ini` and set the entry for `MouseSendsControlV` to `False`. This setting is helpful if you are used to copy/paste by selecting text and then clicking the mouse wheel on the target text entry field. however it is useless if you CAD software makes use of the middle mous click to rotat your Model :)

GPU Forwarding for KVM

Helpful links:

- https://medium.com/@calerogers/gpu-virtualization-with-kvm-qemu-63ca98a6a172
- https://heiko-sieger.info/running-windows-10-on-linux-using-kvm-with-vga-passthrough

```
apt install ovmf
nano /etc/default/grub
# Make this line look like this
GRUB_CMDLINE_LINUX="rdblacklist=nouveau nouveau.modeset=0 intel_iommu=on
iommu=pt rd.driver.pre=vfio-pci video=efifb:off"
# the last video=efifb:off might only be needed if you want to pass through
your primary video controller
# i'm not sure if we need the anti-nouveau stuff here, but i don't like
```

```
nouveau anyway ;)
update-grub
cat >> /etc/initramfs-tools/modules << EOF
vfio
vfio_iommu_type1
vfio_pci
vfio_virqfd
EOF

# get your gpu's vendor and product id:
root@pssrv:~# lspci -nnk | grep -i nvidia
libkmod: ERROR ../libkmod/libkmod-config.c:635 kmod_config_parse:
/etc/modprobe.d/vfio.conf line 1: ignoring bad line starting with 'EOF'
libkmod: ERROR ../libkmod/libkmod-config.c:635 kmod_config_parse:
/etc/modprobe.d/vfio.conf line 2: ignoring bad line starting with 'EOF'
05:00.0 VGA compatible controller [0300]: NVIDIA Corporation GM107GL [Quadro
K1200] [10de:13bc] (rev a2)
    Subsystem: NVIDIA Corporation GM107GL [Quadro K1200] [10de:1140]
    Kernel modules: nvidiafb, nouveau
05:00.1 Audio device [0403]: NVIDIA Corporation Device [10de:0fbc] (rev a1)
    Subsystem: NVIDIA Corporation Device [10de:1140]


cat >> /etc/modprobe.d/vfio.conf << EOF
options vfio-pci ids=10de:13bc,10de:0fbc,10de:1140
options vfio-pci disable_vga=1
EOF

cat >> /etc/modprobe.d/blacklist-framebuffer.conf << EOF
blacklist nouveau
blacklist nvidiafb
EOF

update-initramfs -u

reboot
lspci -nnk -d 10de:13bc
#should show vfio-pci in use:
05:00.0 VGA compatible controller [0300]: NVIDIA Corporation GM107GL [Quadro
K1200] [10de:13bc] (rev a2)
    Subsystem: NVIDIA Corporation GM107GL [Quadro K1200] [10de:1140]
    Kernel driver in use: vfio-pci
    Kernel modules: nvidiafb, nouveau

cd /bulk/vm
qemu-img create -f raw /bulk/vm/windows10.img 100G
cp /usr/share/OVMF/OVMF_VARS.fd /bulk/vm/OVMF_VARS.windows10.fd

qemu-system-x86_64 -enable-kvm -k de-ch -name windows -drive
file=/bulk/vm/windows10.img,if=virtio,format=raw,index=0 -m 6000 -device
virtio-net-pci,netdev=net0,mac=52:54:00:01:00:01 -netdev tap,id=net0 -cdrom
```

```
Win10_1803_English_x64.iso -drive file=virtio-win.iso,media=cdrom,index=1 -
monitor tcp:127.0.0.1:5801,server,nowait -boot once=d -cpu host,kvm=off -smp
cpus=8,cores=8 -device vfio-pci,host=05:00.0 -monitor vc -drive
if=pflash,format=raw,readonly,file=/usr/share/OVMF/OVMF_CODE.fd -drive
if=pflash,format=raw,file=/bulk/vm/OVMF_VARS.windows10.fd -rtc
base=localtime -balloon none -serial none -parallel none -M q35 -vnc :2
```

install windows now the nvidia card should be listed besides the standard vga adapter provided by kvm. nvidia drivers did install automatically on my windows 10 installation. With the K1200 i had to go to the nvidia settings and add a custom resolution to get full 4k out of it.

- http://realtech-vr.com/admin/glview to show the opengl version supported by the windows installation –> Version 1.1.0 what qemu provides.
- `dxdiag` to show info about directX support

2nd Try: HP RGS

HP has a Software Product called RGS (Remote Graphics Server) a customer of mine was using this for 3d rendering on remote desktops and was pretty happy with it. When you look at the bottom of the linked product page, you can actually get a price for a license in their online shop where you see, that it is pretty affordable for a single server license. it costs just short of $300. As far as I understand, no additional servers and such are needed, only the windows machine you want to share and of course some client. They provide Servers for Windows and Linux and the clients for Windows, Mac and Linux. For about $300 you get a single server license (server being the machine providing the desktop you want to conenct to). there is also a floating license available for about twice as much which can be used with several servers and is licensed upon concurrent use. In my case, a single server license is enough.

Luckily they provide a free 60 day trial also at the bottom of the webpage, so lets get started :)

first we copy our windows 10 test VM so we don't mess up the citrix install:

```
cp -p windows10.img windows10HPrgs.img
```

now let's boot it for the first time without the gpu installed, so we can make sure it boots all right:

```
  qemu-system-x86_64 -enable-kvm -k de-ch -name windows -drive
file=/bulk/vm/windows10HPrgs.img,if=virtio,format=raw,index=0 -m 9000 -
device virtio-net-pci,netdev=net0,mac=52:54:00:01:00:01 -netdev tap,id=net0
-cdrom Win10_1803_English_x64.iso -drive file=virtio-
win.iso,media=cdrom,index=1 -monitor tcp:127.0.0.1:5801,server,nowait -boot
once=d -cpu host,kvm=off -smp cpus=8,cores=8 -monitor vc -drive
if=pflash,format=raw,readonly,file=/usr/share/OVMF/OVMF_CODE.fd -drive
if=pflash,format=raw,file=/bulk/vm/OVMF_VARS.windows10.fd -rtc
base=localtime -balloon none -serial none -parallel none -M q35 -vnc :2
```

during this first session I also uninstalled the Citrix VDA stuff, since this probably requires a reboot anyway

now let's reboot with the gpu attached, so we can get to business

```
qemu-system-x86_64 -enable-kvm -k de-ch -name windows -drive
file=/bulk/vm/windows10HPrgs.img,if=virtio,format=raw,index=0 -m 9000 -
device virtio-net-pci,netdev=net0,mac=52:54:00:01:00:01 -netdev tap,id=net0
-cdrom Win10_1803_English_x64.iso -drive file=virtio-
win.iso,media=cdrom,index=1 -monitor tcp:127.0.0.1:5801,server,nowait -boot
once=d -cpu host,kvm=off -smp cpus=8,cores=8 -device vfio-pci,host=05:00.0 -
monitor vc -drive
if=pflash,format=raw,readonly,file=/usr/share/OVMF/OVMF_CODE.fd -drive
if=pflash,format=raw,file=/bulk/vm/OVMF_VARS.windows10.fd -rtc
base=localtime -balloon none -serial none -parallel none -M q35 -vnc :2
```

install the RGS sender and also the virtual mouse stuff

to install the client on my ubuntu desktop sadly the ubuntu directory contains an rpm file with a script that executes an `rpm -Uvh` to install this package.. well, that won't work on ubuntu!

so instead, let's use alien to convert the rpm and install it:

```
sudo apt install alien
sudo alien rgreceiver*.rpm
sudo dpkg -i rgreceiver*.deb
```

now to start rgreceiver we need to set the LD_LIBRARY_PATH and then start it:

```
export LD_LIBRARY_PATH=/opt/hpremote/rgreceiver/lib64:$LD_LIBRARY_PATH
/opt/hpremote/rgreceiver/rgreceiver
```

to launch rgreceiver more conveniently by just typing `rgreceiver` in a new shell, let's add the LD_LIBRARY_PATH export and PATH export to our `.bashrc` to automatically execute each time you open a new shell:

```
echo 'export
LD_LIBRARY_PATH=/opt/hpremote/rgreceiver/lib64:$LD_LIBRARY_PATH' >>
~/.bashrc
echo 'export PATH=/opt/hpremote/rgreceiver:$PATH' >> ~/.bashrc
```

in the connection window that appears after starting the rgreceiver, you can click on settings and select to `match receiver resolution` which will start rgreceiver in fullscreen mode. to get out of fullscreen mode press `leftCtrl + space` and release space while keeping `leftCtrl` pressed, now press H and you will see a little menu at the top of the screen. to enter settings press M instead of H

connecting through a firewall

to enable access to your RDG service you need to allow both TCP and UDP traffic to Port 42966 to be routed to your RDG sender.

First Impression

the performance is similar to Citrix.. the quality, lag and framerate seem very similar to Citrix, I'd say they are about the same. Installation is ALOT easier than citrix, the whole server infrastructure around it is not needed. Therefore i recon this setup will also run more stable in an environment like mine (not talking about big enterprise installs). price is okay as well. the big Downside is: it seems to be ment for fullscreen use only! the desktop size won't scale down if the receiver is run within a window and i haven't found a way yet to have the receiver in windowed mode without having scrollbars around it and needing to scroll. so that's definately a major downside of this solution.

3rd Try: Nice DCV

Nice DCV is our third contestant. Priced at 300€ for a perpetual license this is the most expensive of the bunch. It needs a server component installed but it does not require any installation on the client side anymore.. it used to be RealVNC based but now they have moved to an HTML 5 solution which is nice as far as accessabiliy is concerned. The downside of this is, that HTML 5 clients are a lot more restricted for example when it comes to syncing your Clipboard. DCV requires you to press a button on their control panel to copy your local clipboard to the remote machine and vice versa.. To overcome that, there is a client that can be installed on workstations you use frequently, so you can enjoy both the comfort of a native client and the portability of a web based client which is absolutely awesome. well done!

compared to the HP RGS, DCV supports on-the-fly desktop scaling, so when it is used in windowed mode on the native client or in a browser window, the desktop will resize as soon as you resize the viewer window.

upon initial install you can use the software for 15 days free of chare to try it out. if you need more time, contact sales and they will give you a demo license that will allow you to try some more.

so let's make another copy of our windows 10 image.. this time with rsync so we see the progress :)

```
rsync --progress windows10HPrgs.img windows10NiceDCV.img
```

start it again withou the GPU attached to it for the first time:

```
qemu-system-x86_64 -enable-kvm -k de-ch -name windows -drive
file=/bulk/vm/windows10NiceDCV.img,if=virtio,format=raw,index=0 -m 9000 -
device virtio-net-pci,netdev=net0,mac=52:54:00:01:00:01 -netdev tap,id=net0
-cdrom Win10_1803_English_x64.iso -drive file=virtio-
win.iso,media=cdrom,index=1 -monitor tcp:127.0.0.1:5801,server,nowait -boot
once=d -cpu host,kvm=off -smp cpus=8,cores=8 -device vfio-pci,host=05:00.0 -
monitor vc -drive
if=pflash,format=raw,readonly,file=/usr/share/OVMF/OVMF_CODE.fd -drive
if=pflash,format=raw,file=/bulk/vm/OVMF_VARS.windows10.fd -rtc
base=localtime -balloon none -serial none -parallel none -M q35 -vnc :2
```

download the latest dcv server and install it on the vm, then connect to https://vm-Ip:8443

if you want to use the client, simply download and install the package that matches your client os.

so far i could only test this from a very slow connection in a hotel abroad and it is amazing how much faster both DCV and HP RGS are compared to normal RDP. DCV seems to perform very similar or maybe even a bit better than RGS with this extremely low bandwidth (too low really to use any sort of remote desktop with full HD, not to mention 4k).. so at the moment, except for the high priche tag, DCV is in my eyes the winner so far.

retry DCV

dcv was running very unreliably.. it crashed often when i went to full screen with 4k and it had performance problems with resolutions in the 4k range.. i noticed however, that my method of copying the windows image and then first uninstall the HP stuff before i installed DCV was probably not the best as for example the display driver for the virtual display was not called DCV Display but something else, so i think there might have been some leftovers around which interfere with DCV.. so in order to get rid of all that i will retry the dcv installation with a new windows install. also i noticed that the performance of my virtual disk was lousy (10MB/s) so i am now using an LVM volume on a physical disk rather than an image on a zfs storage.

here is the command line to start the vm:

```
qemu-system-x86_64 -enable-kvm -k de-ch -name windows -drive
format=raw,file=/dev/vms/dcv,if=virtio,format=raw,index=0 -m 9000 -device
virtio-net-pci,netdev=net0,mac=52:54:00:01:00:01 -netdev tap,id=net0 -cdrom
Win10_1803_English_x64.iso -drive file=virtio-win.iso,media=cdrom,index=1 -
monitor tcp:127.0.0.1:5801,server,nowait -boot once=d -cpu host,kvm=off -smp
cpus=8,cores=8 -device vfio-pci,host=05:00.0 -monitor vc -drive
if=pflash,format=raw,readonly,file=/usr/share/OVMF/OVMF_CODE.fd -drive
if=pflash,format=raw,file=/bulk/vm/OVMF_VARS.windows10.fd -rtc
base=localtime -balloon none -serial none -parallel none -M q35 -vnc :1
```

as far as speed goes, this worked out a lot better, however i could not get DCV to run reliably.

this time i've created a snapshot prior to installing DCV and i later cloned that snapshot to install RGS on a “plain” clone of the windows install in a second LVM volume. Here is my command to start the RGS VM:

```
qemu-system-x86_64 -enable-kvm -k de-ch -name windows -drive
format=raw,file=/dev/vms/rgs,if=virtio,format=raw,index=0 -m 9000 -device
virtio-net-pci,netdev=net0,mac=52:54:00:01:00:01 -netdev tap,id=net0 -cdrom
Win10_1803_English_x64.iso -drive file=virtio-win.iso,media=cdrom,index=1 -
monitor tcp:127.0.0.1:5801,server,nowait -boot once=d -cpu host,kvm=off -smp
cpus=8,cores=8 -monitor vc -drive
if=pflash,format=raw,readonly,file=/usr/share/OVMF/OVMF_CODE.fd -drive
if=pflash,format=raw,file=/bulk/vm/OVMF_VARS.windows10.fd -rtc
base=localtime -balloon none -serial none -parallel none -M q35 -vnc :1 -
device vfio-pci,host=05:00.0
```

RGS runs smooth and windows feels reasonably fast too with the SSD as storage behind it. however,

When the vm is left running over night usually the next morning it hangs or is so slow that one often can't even shut it down properly. I don't know why that is, i've deactivated all power saving features of windows, screen blanking etc. but I still have this issue. The machine hangs the same when I connect vie windows RDP. Also in those cases where i can still see something, I usually get alot of horizontal lines as artefacts on the screen both with RGS and RDP. The only solution is then to shut down the VM and start it again, a Reboot (instead of a power cycle) will make the VM fast again but the Artefacts remain. so there is definately something wrong with my VM here. This needs further investigation!

LVM stuff

just as a quick side note: here is how you create and revert to a snapshot in lvm. using dd you can also copy a snapshot into a new lvm volume if the new volume is at least the size of the snapshotted one.

create a snapshot:

```
lvcreate -s -L 50G -n win10base /dev/vms/dcv
```

revert to a snapshot:

```
lvconvert --merge /dev/vms/win10base
```

using a vm with just the forwarded GPU

this is the ultimate goal. so far i could not get this to work, whenver i booted a vm without a emulated graphics card i could no longer connect to it. and it was hard to debug as i had no “local” screen to connect to. however, i recently connected a monitor to the GPU and found out, that the reason why i could not connect was in fact very simple: once the emulated graphics adapter is no longer presence, the pci-id of the emulated NIC changes and windows detects it as a new ethernet adapter despite it having the same MAC. So the vm actually got a DHCP assigned ip rather than the fixed one i had configured. I could then connect to the dhcp assigned ip and set the same fixed ip on the “second” interface which now works fine.

to start a vm without a emulated graphics card, use `-nographic -vga none` in your command line.

From:
http://wiki.psuter.ch/ - **pswiki**

Permanent link:
http://wiki.psuter.ch/doku.php?id=gpu_accelerated_vm_with_kvm&rev=1543444232

Last update: **28.11.2018 23:30**