

Srednja elektro-računalniška šola Maribor

DELOVNA POSTAJA V OBLAKU

Raziskovalno področje: Računalništvo in telekomunikacije

Raziskovalna naloga

Avtorja: Primož Pokeržnik in Žak Karo

Mentor: Branko Potisk

Maribor, januar 2020

KAZALO VSEBINE

POVZETEK	5
1 UVOD.....	6
1.1 OPREDELITEV PROBLEMA	6
1.2 VIRTUALIZACIJA RAČUNALNIKA/STREŽNIKA	6
1.2.1 KAJ JE VIRTUALIZCIJA.....	6
1.2.2 KAKO POTEKA NAVIDEZNO OKOLJE (VIRTUALIZACIJA).....	7
1.3 ZAKAJ UPORABITI STREŽNIK	8
1.4 KAJ JE NA TEM PODROČJU ŽE ZNANEGA	8
2 HIPOTEZE IN CILJI.....	9
3 RAZISKOVANJE	10
3.1 PRIPRAVA STREŽNIKA ZA PRIČETEK RAZISKOVANJA.....	10
3.1.1 KARAKTERISTIKE SISTEMA	10
3.1.2 PRIPRAVA SISTEMA.....	10
3.2 KREIRANJE NAVIDEZNE NAPRAVE.....	12
3.2.1 PRIPRAVLJANJE PROGRAMSKE OPREME	12
3.2.2 PREDELAVA GRAFIČNE KARTICE	13
3.3 POVEZAVA NA NAVIDEZNO DELOVNO POSTAJO	17
3.4 PRIPRAVLJANJE ODDALJENEGA NAMIZJA.....	18
3.5 TESTIRANJE	18
3.5.1 TESTIRANJE V PROGRAMSKEM OKOLJU WINDOWS.....	18
3.5.2 TESTIRANJE V PROGRAMSKEM OKOLJU LINUX	19
3.5.3 PRIMERJAVA PRENOSNIKA Z NAVIDEZNO NAPLAVO.....	19
3.6 NAVIDEZNE NAPRAVE ZA DIJAKE ŠOLE	20
3.7 IMPLEMENTACIJA OBLAKA.....	20
4 DISKUSIJA.....	22
5 ZAKLJUČEK.....	23
6 DRUŽBENA ODGOVORNOST	24
6 VIRI	25

KAZALO SLIK

Slika 1: Prikaz delovanja virtualizacije	6
Slika 2: Nadzorna plošča Proxmoxa	10
Slika 3: Kreiranje mosta	11
Slika 4: Nov most	11
Slika 5: Končne nastavitve mosta	11
Slika 6: Omrežne nastavitve na kontejnerju	12
Slika 7: Dokaz, da strežnik prepozna grafično kartico	12
Slika 8: Napaka 43	13
Slika 9: Zamenjava uporov na grafični kartici	14
Slika 10: Zamenjava upornikov	14
Slika 11: Primer izpisa upravitelja naprav	15
Slika 12: Prikaz ID številke grafičnih čipov	16
Slika 13: »Classroom« test v Blenderju	19
Slika 14: Preizkušanje programa Blender	20

ZAHVALA

Zahvaljujemo se našemu profesorju, ki nam je pomagal pri raziskovanju, in šoli, ki nam je priskrbela potrebno opremo.

Zahvaljujemo se tudi lektorici za lektoriranje naše naloge.

Hvala tudi staršem, ki so nas podpirali pri raziskovanju.

POVZETEK

Raziskovalna naloga predstavlja enega od mogočih načinov virtualizacije delovne postaje. Iskali smo optimalno rešitev za razpoložljivo strojno opremo, za znani problem. In ta je; nekateri dijaki naše šole nimajo dovolj zmogljivih računalnikov za delo domačih nalog pri pouku multimedije. Prišli smo do rešitve, kako bi jim lahko s pomočjo virtualizacije strežnika in navidezne delovne postaje to omogočili.

Pri raziskavi smo posnemali programsko in strojno opremo delovne postaje, ter jo interpretirali na strežniku, kjer se bo izvajala navidezno.

Poleg tega pa bomo na strežniku poganjali še oblačne storitve, npr. lahko bodo prenašali datoteke, s katerimi bodo delali na navideznih delovnih postajah.

1 UVOD

1.1 Opredelitev problema

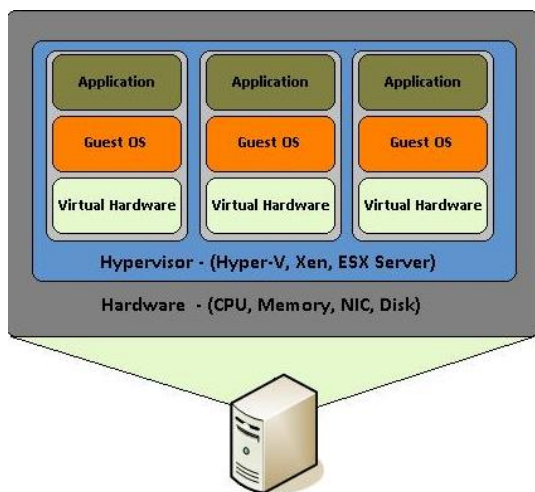
Na šoli imamo predmete kot so npr. multimedija, kjer delamo s programi, ki zahtevajo zmogljivejši računalnik. Pri pouku delamo z delovnimi postajami, katere so sestavljene iz komponent, ki so specifično pripravljene za obdelavo grafik. Takšni računalniki so praviloma precej dražji.

Vsi dijaki doma nimajo primerljivo zmogljivih računalnikov kot jih imamo na šoli, zato vse naloge s področja multimedije, kot so obdelava videa, delajo le v šoli.

Tako smo prišli na idejo, da bi lahko dijakom ponudili navidezne delovne postaje na strežniku (v nadaljevanju navidezne naprave), na katerih bi lahko delali domače naloge. S programom za oddaljeno namizje bi se lahko prijavi iz domačega računalnika, pri katerem ni pomembna njegova zmogljivost.

1.2 Virtualizacija računalnika/strežnika

1.2.1 Kaj je virtualizacija



Slika 1: Prikaz delovanja virtualizacije

Virtualizacija je tehnika, s katero lahko navidezno poganjamo več navideznih računalnikov, »strojev« (angl. virtual machine, »VM«) na enem računalniku. S to tehniko se največkrat srečamo pri strežniških storitvah, saj je njihov glavni namen to da ne potrebujemo več fizičnih strežnikov, na katerih bi poganjali najrazličnejše spletne storitve (strežniki za spletišča, spletno pošto, oblaki...).

Poznamo več vrst virtualizacije: virtualizacija strežnika, aplikacije, namizja, strojne opreme, omrežja, podatkov...

Navidezne naprave so ločene od gostiteljskega operacijskega sistema (angl. hypervisor), kar pomeni, da če je prišlo do zrušitve sistema navidezne naprave, gostujoči operacijski sistem deluje nemoteno.

Najbolj znani gostitelji navideznih naprav so: VMware vSphere, Microsoft Hyper-V, Proxmox, LimeTech UNRAID, XenServer, oVirt...

Navidezne naprave lahko poganjamo neposredno na strežniku (z uporabo gostiteljskega operacijskega sistema), ali preko Windowsa, kjer imamo na voljo več različnih programov, ki simulirajo gostiteljski sistem za virtualizacijo; to so npr. VMware Workstation Player, VirtualBox, Microsoft Hyper-V...

Njihova najpomembnejša značilnost je, da lahko na enem strežniku poganjamo več navidezni naprav (VM), ki delujejo druga od druge neodvisno. Vsaki navidezni napravi lahko določimo delež strojne opreme, ki ga ima na voljo za svoje delovanje, npr. število procesorskih jeder, velikost dodeljenega pomnilnika, podatkovne shrambe (diska), mrežno kartico (navidezno ali fizično), PCI kartice...

Pri raziskavi smo preizkusili tri različne gostiteljske sisteme: VMware ESXi, oVirt in Proxmox. Najprej smo začeli z ESXi, na katerem smo uspeli virtualizirati grafično kartico, vendar smo imeli težave z napako 43 (več o tej napaki v nadaljevanju). Naleteli smo tudi na problem, saj nismo mogli narediti navideznega notranjega NAT-a, ki bi nam pomagal pri problemu z uporabo samo enega javnega IP naslova.

Nato smo preizkusili sistem oVirt. Pri njem nam je uspelo nastaviti interni NAT, vendar smo imeli težave pri virtualiziranju grafične kartice.

Na koncu smo prišli do Proxmoxa, kjer nam je uspelo nastaviti interni NAT ter pravilno virtualizirati grafično kartico. To je bilo ključno, da smo se potem lotili virtualizacije s sistemom Proxmox.

1.2.2 Kako poteka navidezno okolje (virtualizacija)

Za navidezno okolje najprej potrebujemo zmogljiv računalnik (strežnik), ki dovoli poganjanje navidezni naprav. Največkrat je pri tem odločilen procesor, ki je ključnega pomena. Intelovi procesorji imajo to funkcionalnost poimenovano pod kratico VT-x (osnovno virtualiziranje) in VT-d (virtualiziranje ostalih vhodno/izhodnih komponent, npr. grafične kartice). Pri AMD procesorjih to poznamo pod oznako AMD-V. Če ima procesor te oznake, smo na pravi poti do navideznega okolja. Prav tako igra ključno vlogo diskovno polje. Potrebujemo dovolj podatkovnega prostora za vse navidezne naprave, tj. vsaj 20 GB za vsako zmerno veliko navidezno napravo. Najbolje je, da so diski v RAID (angl. Redundant Array of Independent Disks) polju (ni nujno, vendar je zaželeno). To je način združevanja več diskov, v različne skupine. Te skupine imajo različne funkcionalnosti, katere so označene s številko. Z RAID-om lahko pridobimo na večji varnosti podatkov (RAID 1 - kloniranje diskov; najbolj uporabno pri dveh diskih), večji zmogljivosti (RAID 0 - mala varnost podatkov; več skupnega prostora ter

zmogljivosti), oboje hkrati (npr. RAID 5), vendar za to potrebujemo več diskov. Pri poganjanju navideznih naprav to igra veliko vlogo pri zmogljivosti samega gostiteljskega sistema ter navideznih napravah. Če imamo več diskov iste kapacitete, je najbolje da jih združimo v polje RAID 5 ali 6. Če imamo na voljo le dva diska, ju združimo v RAID 0 če želimo ob enem maksimalno zmogljivost in kapaciteto, vendar riskiramo fizično napako diska, ali RAID kartice. Če potrebujemo večjo varnost podatkov, izberemo polje RAID 1, ki bo kloniralo oba diska. Če se en disk pokvari, je takoj na voljo drugi disk.

1.3 Zakaj uporabiti strežnik

Želeli smo, da se lahko na eno napravo poveže več uporabnikov. Strežniki so namenjeni virtualizaciji, pri čemer so zelo fleksibilni. Za razliko od običajnih namiznih računalnikov so namenjeni nenehnemu delovanju. Prav tako so daljšemu obratovanju namenjene komponente strežnika. Med drugim ima večina strežnikov dva procesorja, ki ju lahko odlično virtualiziramo in porazdelimo po različnih navideznih napravah.

1.4 Kaj je na tem področju že znanega

Številna podjetja se ukvarjajo s tem, da dajejo v najem strežnike, na katerih se lahko izvajajo grafična dela. Vendar, ker večina teh storitev ne poganja operacijski sistem Windows, lahko na teh strežnikih poganjamo samo programe, ki jih je določilo podjetje. Večina podjetij daje v najem »igralne postaje«, to so specifični sistemi za igranje računalniških iger. Imajo enak koncept, torej se lahko na njih igra samo igre, ki jih je določilo podjetje.

Mi želimo doseči, da na navidezni delovni postaji izdelujemo grafična dela, vendar z razliko, da lahko končni uporabnik dela na operacijskem sistemu Windows.

2 HIPOTEZE IN CILJI

Hipoteze:

- postavili bomo lahko lasten sistem navidezne delovne postaje, ki bo omogočala grafične operacije – delo z grafikami,
- dijaki se bi lahko od doma povezali na navidezno delovno postajo, na kateri bi lahko gladko, brez problemov delali domače naloge iz multimedijskega pouka,
- datoteke bi lahko prenašali med navidezno napravo in svojim domačim računalnikom preko datotečnega oblaka, katerega bi lahko poganjali na istem strežniku.

Cilji:

- postaviti lastni sistem delovnih postaj,
- da je na navideznih delovnih postajah nameščena licenca za Adobe programe za dijake,
- dijaki ne bi potrebovali svojih licenc za domačo uporabo, saj bi bile te nameščene na strežniku.

3 RAZISKOVANJE

3.1 Priprava strežnika za pričetek raziskovanja

3.1.1 Karakteristike sistema

Pri raziskavi smo uporabili enega izmed šolskih strežnikov, ki je bil prost in pripravljen na preizkušanje s sledečimi specifikacijami:

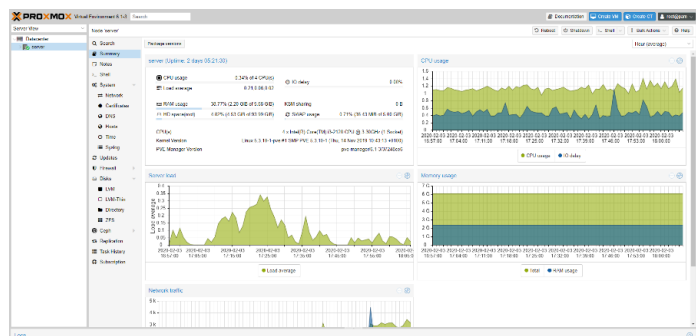
- CPE: 2x Intel Xeon E5-2680 v1 (2,7 GHz – 3,5 GHz, 8 jeder in 16 niti na procesor),
- RAM: 16x 16GB DDR3 1866MHz ECC (skupno 128GB),
- Diski: 4x 3TB SATA 7,2k RPM (RAID 5),
- GPE: Nvidia Tesla K10 (skupno 8GB vRAM).

3.1.2 Priprava sistema

Po prejemu strežnika smo najprej hoteli postaviti diske v ZFS RAID (programski RAID, angl. software RAID). RAID kartico smo najprej želeli postaviti v JBOD način, kar pomeni, da vse diske predstavi kot individualne diske, saj če bi vsak disk posebej postavili v RAID 0, bi izgubili S.M.A.R.T. funkcionalnost (sistem za nadziranje diskov, ter njihovo diagnostiko). Ker pa naša RAID kartica tega uradno ne podpira, smo se odločili za strojni RAID, saj smo bili vsi mnenja, da je strojni RAID bolj varen od programskega. Prav tako imamo dostop do iLo vmesnika (ločeni nadzornik strežnika za oddaljeno nadzorovanje), kjer lahko vidimo »zdravje« in stanje diskov.

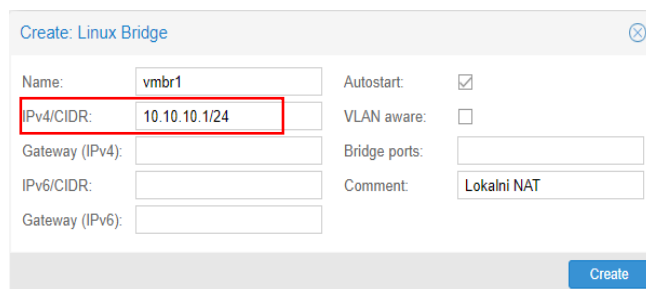
Ko smo končali z nastavljanjem RAID-a, smo se lotili namestitve gostitelja Proxmox. Zanj smo se odločili, saj temelji na Linux Debian-u, kar pomeni, da je precej fleksibilen in stabilen sistem, kateri nam je znan. Ima dober spletni uporabniški vmesnik, je sicer malce zahtevnejši za uporabo, vendar ima dosti priročnih funkcij, kot npr. »cluster« z več strežniki, »high availability«... Pri namestitvi smo nastavili skrbniško geslo, ter zunanji IP naslov.

Po namestitvi se povežemo na IP naslov strežnika, kjer se prikaže nadzorna plošča. Opazimo lahko več grafov, ki prikazujejo obremenitev strežnika.



Slika 2: Nadzorna plošča Proxmoxa (vir: avtorji)

Ker smo dobili le en zunanji IP, smo se lotili kreiranja lokalnega navideznega NAT-a. Najprej smo ustvarili nov Linux Bridge. To lahko naredimo na server → System → Network → Create Linux Bridge.



»Mostu« smo nastavili naslednje lastnosti:

- ime: vmbr1
- IPv4/CIDR: 10.10.10.1/24

Sledil je ponovni zagon sistema.

Slika 3: Kreiranje mosta (vir: avtorji)

```
auto vmbr1
iface vmbr1 inet static
    address 10.10.10.1
    netmask 255.255.255.0
    bridge-ports none
    bridge-stp off
    bridge-fd 0
```

Po prijavi nazaj na strežnik, smo odprli SSH povezavo (oddaljeno ukazno okolje) na Proxmox, ter odprli datoteko /etc/network/interfaces. Tukaj lahko vidimo nov most (glej sliko 4).

Slika 4: Nov most (vir: avtorji)

Temu smo dodali tri linije kode (glej sliko 5), katere bodo vse IP naslove na podomrežju 10.10.10.0/24 preusmerile na glavni priključek eth1 (masquerade in ip_forward).

```
auto vmbr1
iface vmbr1 inet static
    address 10.10.10.1
    netmask 255.255.255.0
    bridge-ports none
    bridge-stp off
    bridge-fd 0
# Virtual NAT
post-up echo 1 > /proc/sys/net/ipv4/ip_forward
post-up iptables -t nat -A POSTROUTING -s '10.10.10.0/24' -o eth1 -j MASQUERADE
post-down iptables -t nat -D POSTROUTING -s '10.10.10.0/24' -o eth1 -j MASQUERADE
```

Slika 5: Končne nastavitve mosta (vir: avtorji)

Sistem smo po končanem nastavljanju še enkrat ponovno zagnali, da se spremembe uveljavijo. Ko se je sistem vzpostavil, smo kreirali Proxmoxov kontejner (angl. container, izoliran Linux sistem, ločen od gostitelja) z Linux Ubuntu-jem. Dodelili smo mu 1 procesorsko jedro, 512MB pomnilnika, ter 4GB diskovnega prostora. Ta kontejner je namenjen samo testiranju, zato smo mu tudi dodelili male kapacitete. Kar pa je najpomembneje je to, da smo mu nastavili statični IP naslov na novem mostu.

Create LXC Container ✕
 General Template Root Disk CPU Memory **Network** DNS Confirm
 Name: eth0 IPv4: Static DHCP
 MAC address: auto IPv4/CIDR: 10.10.10.100/24
 Bridge: vbr1 Gateway (IPv4): 10.10.10.1
 VLAN Tag: no VLAN IPv6: Static DHCP SLAAC
 Rate limit (MB/s): unlimited IPv6/CIDR:
 Firewall: Gateway (IPv6):

Določili smo most, statični IP kontejnerja in IP glavnih vrat. Požarni zid smo pustili vklopljen.

Slika 6: Omrežne nastavitve na kontejnerju (vir: avtorji)

Kontejner deluje kot mora, s tem smo zaključili nastavljanje in preizkušanje IP naslovov. Prav tako je bil strežnik pripravljen za pričetek raziskave.

Ker bomo za raziskavo potrebovali več grafične moči, smo v strežnik vgradili namensko grafično kartico Nvidia Tesla K10.

3.2 Kreiranje navidezne naprave

3.2.1 Pripravljanje programske opreme

Po vgradnji grafične kartice v strežnik, smo potrdili, da strežnik in Proxmox grafično kartico zaznata. To smo naredili z ukazom »*lspci*« v SSH okolje strežnika (glej sliko 7).

```
04:00.0 3D controller: NVIDIA Corporation GK104GL [Tesla K10] (rev a1)
05:00.0 3D controller: NVIDIA Corporation GK104GL [Tesla K10] (rev a1)
```

Slika 7: Dokaz, da strežnik prepozna grafično kartico (vir: avtorji)

Nadaljevali smo s procesom kreiranja navidezne naprave. Dodelili smo ji 8 navideznih jeder na prvem procesorju. Prav tako smo ji dodelili tudi 32GB pomnilnika (brez »ballooning« opcije), 128GB podatkovnega prostora, ter statični IP naslov na mostu. Odločili smo se, da bomo najprej sistem pripravili do te mere, da bo funkcionalen, torej da bodo nameščeni vsi gonilniki, ter potrebni programi, in šele potem dodelili en čip grafične kartice (saj je grafična kartica sestavljena iz dveh samostojnih čipov).

S pritiskom na gumb »kreiraj« smo odprli na novo ustvarjeno navidezno napravo, prikazalo se nam je tudi pogovorno okno z vsemi podatki, ter nastavitvami.

Odločili smo se za uporabo operacijskega sistema Windows 10 Professional.

Na strežnik smo prenesli Windowsovo ISO datoteko, prav tako pa tudi ISO datoteko s Proxmoxovimi odprtokodnimi gonilniki za navidezne naprave.

Navidezni napravi smo nato dodelili navidezni CD/DVD pogon z ISO datoteko Windowsov, ter jo zagnali. Kot po pričakovanjih smo uspešno pred seboj zagledali zagonski zaslon Windowsov. Namestitev sistema je potekala nemoteno, vendar pa je dobro tudi omeniti, da najprej nismo videli navideznih diskov. Problem smo rešili tako,

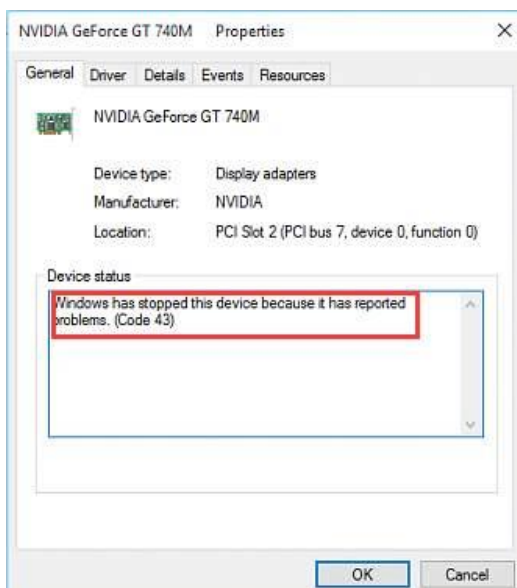
da smo zamenjali ISO datoteko na navideznem CD/DVD pogonu z ISO datoteko z gonilniki, ter jih namestili na inštalaciji. Po uspešni namestitvi gonilnikov, smo lahko mirno ponovno zamenjali ISO datoteko na pogonu, ter nadaljevali z namestitvijo.

Po namestitvi Windowsov smo namestili vse potrebne gonilnike, ter vse potrebne programe: Adobe After Effects, Adobe Premiere Pro in Blender.

Na namizje smo si prav tako pripravili namestitveni program za gonilnike grafične kartice.

Navidezno napravo smo nato zaustavili, ter ji dodelili PCI kartico – grafično kartico. Nato smo jo ponovno zagnali.

Po prijavi v Windowse smo namestili grafične gonilnike, ter ponovno zagnala napravo.

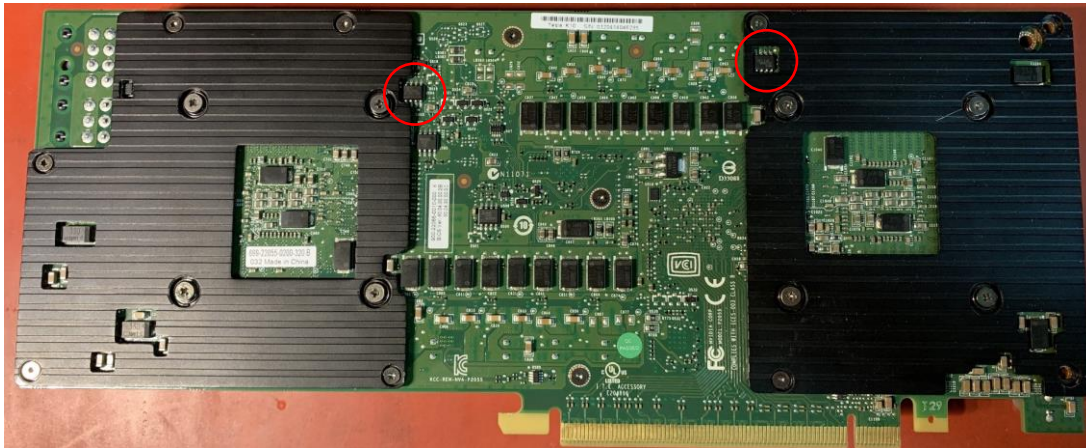


Po prijavi v namizje smo odprli program »upravitelj naprav«, kjer smo bili presenečeni, da grafični gonilniki niso delovali. Ko smo pregledali dnevniške zapise gonilnika, smo naleteli na napako »43« (slika 8). Po hitrem »googlanju« napake smo ugotovili, da Nvidia ne dovoli virtualiziranja njihovih potrošniških grafičnih kartic, temveč nas tako prisili, da uporabimo njihove specialne strežniške kartice, ki so namenjene uporabi CUDA jeder, to so kartice z imenom GRID.

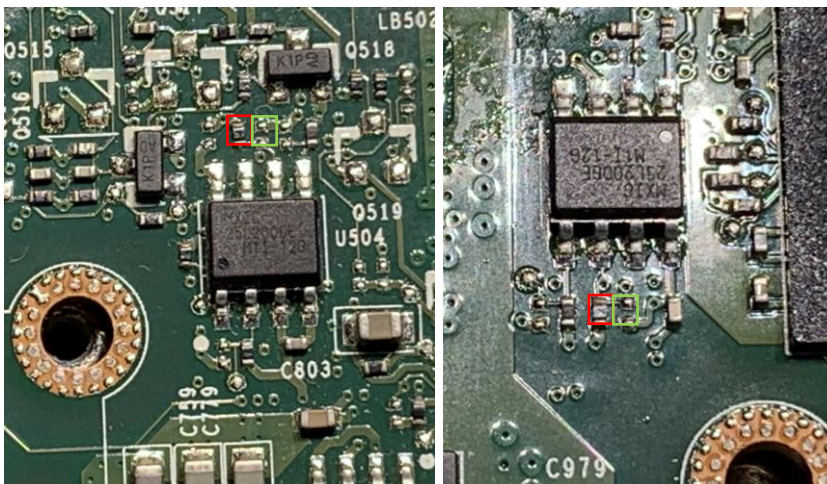
Slika 8: Napaka 43

3.2.2 Predelava grafične kartice

Ker Nvidia podpira uporabo navideznih naprav le na GRID grafičnih karticah, nismo mogli več nadaljevati. Zato smo potem pregledali EEVBLOG-ov forum, kjer so zapisali, da lahko Teslo K10 z zamenjavo uporov, »preobrazimo« v GTX 690, Quadro K5000, ter Grid K2.



Slika 9: Zamenjava uporov na grafični kartici (vir: avtorji)



Slika 10: Zamenjava upornikov (vir: avtorji)

Na pozicijah označenih z rdečo barvo smo oba upora odstranili, na pozicijah označenih z zeleno barvo pa smo pri spajkali upornika 40kΩ.

Nato smo grafično kartico vstavili v drug namizni računalnik (z nameščenimi grafičnimi gonilniki). Ob zagonu računalnika v Windowse, smo se najprej napotili v program upravitelj naprav, kjer smo poleg integriranje grafike opazili prve znake delovanja »modifikacije«. Program je uspešno prepoznal grafično kartico kot GRID K2. Vendar to še ni vse, BIOS na grafični kartici še vedno misli, da sta čipa od Tesle, zato smo se napotili k nalaganju novega BIOS-a na grafično kartico.

Najprej smo odprli program GPU-Z, kjer smo naredili varnostno kopijo prvotnega vBIOS-a. Nato smo s spleta prenesli vBIOS za GRID K2, ter oba vBIOS-a (obstoječega in novega) pregledali v programu KeplerBiosTweaker-1.27. Oba vBIOS-a sta bila identična, razen imena grafične kartice. Tako smo se prepričali, da bo nov vBIOS

spremenil le ime grafične kartice, ter da ne bo prihajalo konfliktov med vBIOS-om in Windowsom. Lotili smo se nalaganja novega vBIOS-a s pomočjo programa Nvflash. Po uspešnem nalaganju, smo ponovno zagnali sistem, grafična kartica je delovala kot je morala, upravitelj naprav jo je pravilno prepoznal, ter vsi programi (Adobe After Effects, Premiere Pro in Blender) so pravilno, tekoče in brez problema obdelali vse demonstracijske videe.

Odločili smo se, da bomo grafično kartico vgradili nazaj v strežnik. Pričakovali smo, da navidezna naprava ne bo pravilno interpretirala grafično kartico, saj je prejšnja (Tesla) pri gonilniku imela drugo ime in smo predvidevali, da bo prišlo do napak pri gonilnikih. Zato smo se odločili, da bomo celotno navidezno napravo odstranili, ter jo z istimi specifikacijami na novo kreirali po enakem postopku.

Po namestitvi vseh gonilnikov, smo prišli do enakega problema; napaka 43. Pomislili smo, da je to lahko problem samo še v Proxmoxu. Nvidiini gonilniki ne podpirajo navidezni naprav, zato smo se spomnili, da Proxmox navidezni napravi pove, da je navidezna in ne fizična naprava.

Common KVM processor			
Utilization	Speed	Base speed:	2,99 GHz
3%	2,99 GHz	Sockets:	1
Processes	Threads	Handles	Virtual processors: 2
98	1207	46018	Virtual machine: Yes
		L1 cache:	N/A
Up time			
0:04:52:57			

To smo potrdili s tem, da v upravitelju opravil pod zavihkom »CPE« vidimo, da je naprava navidezna (glej sliko levo). Proxmox preko grub-a pove grafični kartici informacije o sistemu; torej navidezni procesor, pomnilnik...

Slika 11: Primer izpisa upravitelja naprav

Na spletu smo našli Proxmox-ov »Wiki« portal, kjer je bilo opisano kako se pravilno ta problem odpravi. Odprli smo SSH povezavo na gostitelja (Proxmox), kjer smo uredili konflikte z virtualiziranjem grafične kartice.

V datoteki `/etc/default/grub` smo poiskali poved, katero bomo spreminjali: »`GRUB_CMDLINE_LINUX_DEFAULT="quiet"`«, ter ji v narekovajih dodali funkcijo: »`intel_iommu=on`«, kar pomeni, da IOMMU napravam (grafični kartici) ne bo posredoval podatkov o sistemu. Ko smo z urejanjem datoteke končali, smo zagnali ukaz, da se je datoteka osvežila: »`update-grub`«.

Tako smo končali urejanje Grub-a. Urediti še moramo VFIO module, kateri nam bodo pomagali pri OMVF-u (podpora UEFI-ja na novejših različicah QEMU-ja¹).

Z urejevalnikom besedil smo odprli datoteko `/etc/modules`, kjer smo na konec dokumenta dodali sledeče module: `»vfi0«`, `»vfi0_iommu_type1«`, `»vfi0_pci«` in `»vfi0_virqfd«`. Ko smo končali, smo shranili datoteko in zaprli urejevalnik besedila.

Zdaj moremo urediti tudi IOMMU prekinitve (angl. interrupt remapping). V terminal Proxmoxa smo napisali sledeča ukaza: `»echo "options vfi0_iommu_type1 allow_unsafe_interrupts=1" > /etc/modprobe.d/iommu_unsafe_interrupts.conf«` in `»echo "options kvm ignore_msrs=1" > /etc/modprobe.d/kvm.conf«`. S to funkcijo gostitelj omogoča kompatibilnost in izolacijo vhodno-izhodnih naprav (npr. grafičnih kartic). S to funkcionalnostjo lahko s programsko opremo določamo specifični program, ki nadzira cilje prekinitev.

Nato smo gostitelju prepovedali uporabo grafične kartice za svoje namene; to smo naredili z ukazom: `»echo "blacklist nvidia" >> /etc/modprobe.d/blacklist.conf«`.

Za konec smo še VFIO-u omogočili uporabo grafične kartice. Najprej smo s pomočjo ukaza `»lspci -v«` preverili identifikacijsko številko strojne opreme grafične kartice (glej sliko 12).

```
04:00.0 3D controller: NVIDIA Corporation GK104GL [Tesla K10] (rev a1)
05:00.0 3D controller: NVIDIA Corporation GK104GL [Tesla K10] (rev a1)
```

Slika 12: Prikaz ID številke grafičnih čipov

S pomočjo ukaza `»lspci -n -s 04:00«` ter `»lspci -n -s 05:00«` smo iz obeh čipov prekopirali ID izdelovalca. Na koncu smo z ukazom `»echo "options vfi0-pci ids=<izdelovalčev ID> disable_vga=1"> /etc/modprobe.d/vfi0.conf«` dodali ID izdelovalca na seznam VFIO. Za konec smo izvedli ukaz `»update-initramfs -u«`, ter ponovno zagnali strežnik.

Nato smo želeli preizkusiti delovanje grafične kartice na navidezni napravi. Zagnali smo navidezno napravo, ter odprli upravitelja naprav. Po pričakovanjih, se je grafični gonilni pravilno namestil, ter Windowsi so pravilno zaznali kartico in niso javili nobenih težav ali napak.

Zdaj smo »končno« bili pripravljeni na raziskovanje.

¹ QEMU je odprtokodni posnemalnik naprav.

3.3 Povezava na navidezno delovno postajo

Za dostop do namizja lahko koristimo program noVNC, ki je že vključen v sistem Proxmox. Njegova značilnost (običajno prednost, vendar v našem primeru slabost) je, da je program optimiziran za počasnejše povezave, kar pomeni, da je namenjen več ali manj za strežniško upravljanje, za kakšne kompleksnejše zadeve pa ni primeren, saj deluje z majhnim številom sličic na sekundo, kar je za delovno postajo velik problem. Naš cilj je, da se želimo čimbolj približati »pravi« delovni postaji, torej da imamo občutek, da je to fizični računalnik, kot pa da je navidezen in oddaljen od nas. Zato smo prebrskali več rešitev za oddaljeno namizje. Najbolj znan program za to je Windowsovo orodje »Oddaljeno namizje«, vendar zanj moramo odpreti vrata na usmerniku, ter tvegamo številne napade na sistem. Zato smo to orodje »prečrtali«.

Naslednji najbolj priljubljen program je TeamViewer. Je brezplačen (do neke mere) in je preprost za uporabo. Program deluje tako, da se na strežnik namesti program imenovan »Host«, kateri generira ID gostitelja in geslo. Odjemalec (angl. client) se poveže na strežnik s pomočjo prej generiranega ID-ja in gesla. Program je preprost za uporabo in je dokaj zanesljiv. Med preizkušanjem ni prišlo do nobenih zapletov, napak ali kakršnihkoli upočasnjevanj. Vendar ker je kot noVNC optimiziran za slabše povezave. TeamViewer je narejen za grafično nezahtevna običajna oddaljena opravila in ni narejen za kakršnokoli urejanje posnetkov ali kakršnegakoli resnega (grafičnega) opravila. Zato smo morali najti drugo rešitev.

Naslednja rešitev je program Steam (spletna trgovina z računalniškimi igrami) in njihov vgrajen »in-home streaming«, preko katerega lahko upravljamo računalnik preko lokalnega omrežja in ne preko prostranega omrežja. Za lokalne potrebe je zadeva funkcionirala popolno. Delovanje je bilo gladko, brez problemov. Za grafična opravila je bil program zelo zanesljiv in deloval nad našimi pričakovanji.

Manj znan je tudi program Parsec, ki je zelo podoben Steam-ovem in-home streaming-u, vendar Parsec deluje med drugim tudi preko zunanjih povezav. Program deluje na dokaj podobni osnovi kot Steam in-home streaming.

Tako smo se na koncu odločili, da bomo uporabljali program Parsec, saj je našim potrebam najbolj zadostoval, med drugim pa smo v nadaljevanju tudi preizkušali Steam-ov program.

3.4 Pripravljanje oddaljenega namizja

Na navidezno napravo smo namestili vse potrebne programe, med drugim tudi Parsec. Registrirali smo si račun, ter omogočili oddaljeno upravljanje. Izključili smo tudi vse omejitve pasovne širine, saj imamo na šoli dobro internetno povezavo. Tukaj želimo še omeniti, da je pri tem zelo pomembna pasovna širina omrežja, na katerem se poganja strežnik, saj ne samo, da potrebuje dovolj zmogljivo strojno opremo za procesiranje slike, temveč tudi dovolj zmogljivo internetno povezavo (priporočljivo je imeti vsaj 50 megabitov na sekundo hitrosti nalaganja, angl. upload).

Naš odjemalec je bil prenosnik na mobilnem omrežju. Izbita omrežja ni bila naključna, saj je bil naš cilj, ustvariti testno okolje, čim podrobnejše predvidenim delovnim razmeram, ko bi se lahko dijaki povezali na navidezne naprave preko domačega računalnika.

Povezava med odjemalcem in strežnikom je delovala odlično, zato smo začeli s testi.

3.5 Testiranje

Ker pogosto prihaja do razlik med operacijskima sistema Windows in Linux pri času izvažanja, smo se odločili to preizkusiti. Windows bi naj imel ogromno procesov, ki se izvajajo v ozadju (posodobitve, protivirusna zaščita...), se dostikrat zgodi, da ti procesi upočasnjujejo izvažanje slik in posnetkov. V okolju Linux je teh procesov manj, med drugim pa jih je preprosto upravljati (ni protivirusne zaščite, ni avtomatičnih posodobitev, ki se izvajajo v ozadju...).

3.5.1 Testiranje v programskem okolju Windows

Najprej smo preizkusili delo s programom Blender. Iz spleta smo prenesli preizkuševalne (angl. demo) datoteke, s katerimi smo pričeli testirati zmožnosti grafične kartice. Pri testiranju smo uporabili preizkus »classroom«, saj najbolje simulira delo uporabnika v Blenderju. Naš cilj je bil, da bo delo preko oddaljenega računalnika tekoče in brez prekinitev. To hipotezo smo potrdili. Uporabniška izkušnja je bila neprecenljiva, program je delal tekoče in brez problemov.

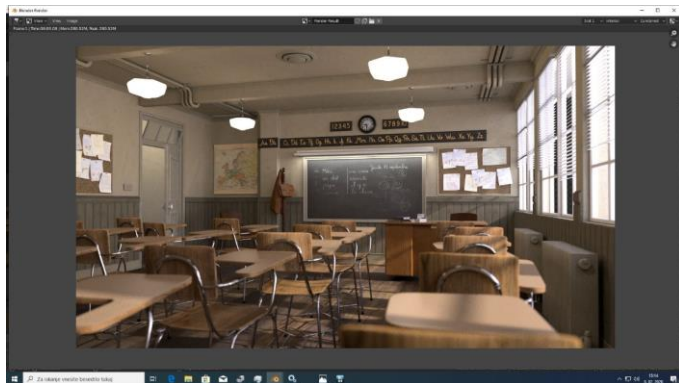
Naslednje kar smo preizkušali, sta bila programa Adobe After Effects in Premiere Pro. Uporabnik je najprej moral zmontirati posnetke v programu Premiere Pro, ter na filmskem traku enega izmed njih urediti v programu After Effects, kjer bo v posnetek vstavil več različnih efektov.

Test se je na koncu izkazal kot uspešnega, saj je celotni sistem deloval tekoče in brez problemov.

3.5.2 Testiranje v programskem okolju Linux

Po končanem testiranju v Windowsih, smo se odločili iste teste opraviti v operacijskem sistemu Linux. Pri testiranju smo uporabili Linux Manjaro Cinnamon (Arch). Kreirali smo navidezno napravo z istimi specifikacijami, ter jo poganjali vzporedno z Windows navidezno napravo, vendar z izjemo, da smo ji dodelili drugi grafični čip, da bo lahko Windows navidezna naprava imela svoj grafični procesor, kot tudi Linux. Tako smo prišli do tega, da lahko na isti grafični kartici poganjamo dva samostojna sistema.

Ker v Linux okolju Adobe programi ne tečejo, smo testirali samo program Blender. Opravili smo enak test kot na Windows navidezni napravi, torej projekt »classroom«.



Izkazalo se je, da bistvene razlike pri izvažanju posnetka med Windowsom in Linuxom ni bilo; Linux je končal par sekund prej, takšna razlika pa je seveda zanemarljiva.

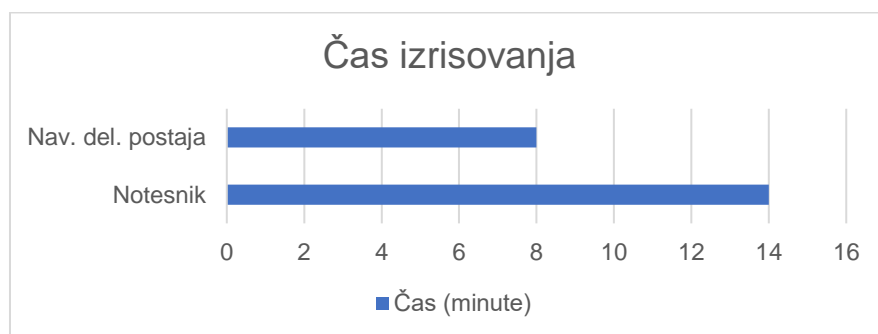
Slika 13: »Classroom« test v Blenderju (vir: avtorji)

3.5.3 Primerjava prenosnika z navidezno napravo

Z navidezno napravo smo tudi primerjali povprečno zmogljiv prenosnik.

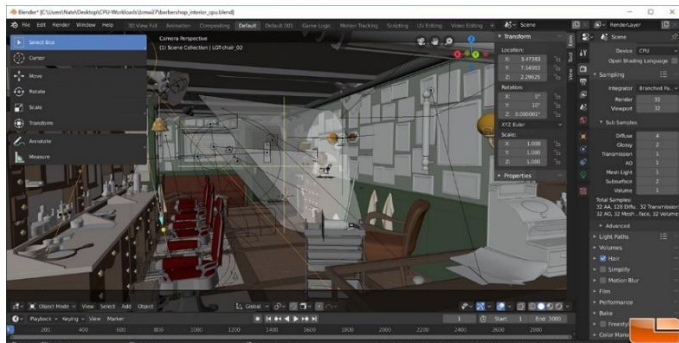
Na obeh sistemih smo izvedli enak preizkus v programu Blender, t.i. »classroom« test. Izbrali smo prenosnik s sledečimi specifikacijami: i5-7200U (CPE), 8GB (pomnilnik), ter Nvidia GeForce 940MX (GPE).

Primerjali smo čas izrisa ene sličice, z ločljivostjo 1280x720.



Grafikon 1: Primerjava časov izrisovanja

Vidimo lahko, da je dokaj zmogljiv prenosnik počasi izrisoval sliko. Naša navidezna delovna postaja bi tako lahko z večjo grafično močjo pripomogla dijakom pri delu z grafikami, prav tako pa tudi pridobimo na času.



Slika 14: Preizkušanje programa Blender

3.6 Navidezne naprave za dijake šole

Ker želimo dijakom, našim sošolcem na šoli omogočiti delo od doma, smo razmišljali, kako to realizirati. Ker lahko na našem testne sistemu naenkrat delata le dva dijaka, smo se dogovorili za termine. Ideja je bila, da je lahko vsak dijak po eno uro na navidezni delovni postaji, potem pa jo preda svojemu sošolcu. Ideja je potekala odlično, dijaki so bili zadovoljni z odzivnostjo in tudi nad tem, da se lahko od doma učijo uporabe Adobe programov in med drugim tudi delajo domače naloge.

3.7 Implementacija oblaka

Ker strežnik skoraj nikoli ni pod polno obremenitvijo, lahko na strežniku poganjamo tudi druge stvari, kot je npr. oblak. Dijaki lahko imajo do njega dostop, tako da lahko pred začetkom dela na oblak od doma naložijo vse potrebne datoteke, ter kasneje ko je za njih na voljo navidezna naprava, bodo imeli vse podatke že na strežniku, tako da lahko takoj pričnejo z delom.

To idejo smo rešili z uvedbo novega kontejnerja, na katerem poganjamo sistem Ubuntu. Nanj smo nato preko storitve »snap« namestili oblak Nextcloud. Oblak smo nato dodali na DNS zapise na šolsko domeno, ter zavarovali povezavo z Let's Encrypt SSL certifikatom. Za konec smo še dijakom kreirali uporabniška imena in gesla, ter na navidezne delovne postaje namestili omrežni pogon, kateri je povezan z oblakom. Med navideznimi napravami in oblakom poteka navidezno stikalo (angl. switch) z 100 gigabitno povezavo, torej ne bo nobenih ozkih grl pri pretoku podatkov.

Prav tako smo vzpostavili povezavo z oblakom in navidezno delovno postajo, ter kreirali omrežni pogon za lažji dostop podatkov na oblaku in delovno postajo.

4 DISKUSIJA

Po končani raziskavi smo razmislili, kako bi lahko v prihodnosti zadevo izboljšali. Prišli smo na več idej:

- ker imamo na voljo samo eno grafično kartico, lahko naenkrat delata samo dve navidezni postaji, zato lahko hkrati delata samo dva dijaka. To bi lahko rešili s tem, da bi imeli na voljo več navideznih naprav, katere bi se vklopile po določenem času. Nastavi bi lahko, da se vklopijo po določenem času, oziroma ko je razpoložljiva strojna oprema,
- v naš trenutni strežnik bi lahko vgradili še eno grafično kartico, tako da bi lahko potem delali štirje dijaki naenkrat,
- lahko bi kupili večji in močnejši strežnik z močnejšimi procesorji, v katerega bi lahko potem vstavili več grafičnih kartic.

Prvo hipotezo, da bi lahko poganjali dve grafični delovni postaji naenkrat smo potrdili, saj nam je to uspelo, strojna oprema je zadovoljila naša pričakovanja (ni bilo ozkih grl). Drugo hipotezo, da se bi lahko dijaki povezali na strežnik in opravljali od doma smo potrdili, saj nam je uspelo povezati Parsec račun z gostiteljem in odjemalcem. Povezava je bila stabilna, tekoča in je potekala brez težav.

Tretjo hipotezo, da bi lahko dijaki prenašali vse datoteke preko oblaka smo potrdili, saj smo tudi to rešili na istem strežniku, kot tretjo navidezno napravo.

Cilj, da bi prihranili pri številu uporabljenih licenc, pa se je izkazal za brezpredmetnega, ker so se pri Adobe odločili, da je licenca vezana na vsakega posameznika.

5 ZAKLJUČEK

Navidezna delovna postaja bi tako lahko omogočila dijakom, kateri nimajo dovolj zmogljivega računalnika možnost, da lahko domače naloge pri multimediji delajo od doma na šolskem strežniku.

Na navidezne delovne postaje bi se lahko dijaki povezali od koderkoli preko programa Parsec. Potrebovali bi le dovolj dobro internetno povezavo, priporočljivo je vsaj 10 megabitov v obe smeri za najbolj optimalno delovanje.

Pri raziskovanju smo naleteli na dosti težav, katere smo uspešno rešili. Največji problem je bila nenamenska grafična kartica, katero smo uspeli uporabiti s tem, da smo zamenjali par uporov in vBIOS.

V nalogi smo podrobno opisali postopke postavitve sistema in upamo s tem tudi vzpodbudili še koga da se loti te naloge.

Naš predlog oziroma ideja je, da bi v prihodnje kupili več strežniških sistemov z grafičnimi karticami in tako omogočili delo več dijakom. V en, v našem primeru uporabljen strežnik lahko damo dve grafični kartici, tako da če bi kupili še tri enake strežniške sisteme, bi lahko naenkrat opravljalo svoje delo 12 dijakov.

6 DRUŽBENA ODGOVORNOST

S postavitvijo virtualiziranih delovnih postaj smo pokazali, da je to mogoče realizirati tudi v šolskem okolju in prinaša kar nekaj prednosti.

Npr.:

Dijaki ne bi rabili kupovati zmogljivejših računalnikov, saj bi lahko vse opravili v šoli ali pa na daljavo na šolskem sistemu.

6 VIRI

Kaj je virtualizacija?, Spletina, dostopno na:

<http://www.spletina.si/internet/virtualizacija.html> [7.2.2020]

What is virtualisation?, Microsoft Azure, dostopno na: <https://azure.microsoft.com/en-in/overview/what-is-virtualization/> [17.1.2020]

PCI passthrough via OVMF, ArchLinux Wiki, dostopno na:

https://wiki.archlinux.org/index.php/PCI_passthrough_via_OVMF [6.2.2020]

QEMU, ArchLinux Wiki, dostopno na: <https://wiki.archlinux.org/index.php/QEMU> [6.2.2020]

Hacking NVidia Cards into their Professional Counterparts, EEVBlogs, dostopno na: <https://www.eevblog.com/forum/general-computing/hacking-nvidia-cards-into-their-professional-counterparts/> [5.1.2020]

Here are 5 types of virtualization, TechAdvisory, dostopno na:

<https://www.techadvisory.org/2018/07/here-are-5-types-of-virtualization/> [5.1.2020]

Program Parsec, dostopno na: <https://parsecgaming.com/> [5.1.2020]

Program Steam, dostopno na: <https://store.steampowered.com/remoteplay> [5.1.2020]

Okolje Proxmox, dostopno na: <https://www.proxmox.com/> [5.1.2020]

Viri slik

Vse slike smo posneli sami, nekaj pa smo jih vzeli iz spleta:

Slika 1: https://op-tec.co.uk/assets/img/Hardware_Virtualisation.jfif [15.1.2020]

Slika 8: <https://images.drivereasy.com/wp-content/uploads/2016/09/nvidia-code-43-error.jpg> [4.2.2020]

Slika 9: <https://cdn.videocardz.com/1/2013/03/GeForce-GTX-690-Into-Quadro-K5000.jpg> [5.2.2020]

Slika 10: <https://i.stack.imgur.com/QYm5Z.png> [6.2.2020]

Slika 14: <https://www.legitreviews.com/wp-content/uploads/2019/09/blender-280.jpg> [6.2.2020]