PD.TVZ.HR
POLYTECHNIC & DESIGN Vol. 13, No. 1, 2025.
Mrežna inačica: ISSN 2459-6302
Tiskana inačica: ISSN 1849-1995
DOI: 10.19279/TVZ.PD.2024-13-1-07

USPOREDBA PERFORMANSI I IMPLEMENTACIJE PROSLJEĐIVANJA GRAFIČKIH KARTICA NA HYPER-V I KVM HIPERVIZORIMA

COMPARISON OF THE PERFORMANCE AND IMPLEMENTATION OF GPU PASSTHROUGH ON HYPER-V AND KVM HYPERVISORS

Nikša Kovačić[1], Ognjen Mitrović[2], Vedran Tadić[2], Tin Antonić[2]

[1]Tehničko veleučilište u Zagrebu, Vrbik 8, Zagreb, Hrvatska, Student
[2]Tehničko veleučilište u Zagrebu, Vrbik 8, Zagreb, Hrvatska

SAŽETAK

Ovaj rad istražuje i uspoređuje mogućnosti prosljeđivanja grafičkih kartica na Hyper-V i KVM hipervizorima prema virtualnim strojevima, detaljno opisujući metode, postupke i razlike u arhitekturi koje utječu na implementaciju. Jedan od bitnih izazova bilo je osigurati provedbu testiranja na istom hardveru, čime su osigurani pouzdani i usporedivi rezultati performansi na različitim operativnim sustavima. Cilj rada je ponuditi platformu za testiranje performansi i kompatibilnosti grafičkih kartica s virtualiziranim aplikacijama na različitim hipervizorima. Također, željelo se eliminirati potrebu za specijaliziranim i skupim poslužiteljskim sustavima te omogućiti testiranje grafičkih kartica u komercijalno dostupnom hardveru. U sklopu praktičnog dijela uključeni su koraci implementacije sustava te je izvršeno i testiranje performansi dostupne grafičke kartice u svim opisanim scenarijima. Time je stvoreno praktično okruženje na jednom fizičkom računalu koje omogućava daljnja istraživanja i replikaciju. Dobiveni rezultati pokazuju kako Hyper-V ima manju prednost što se tiče performansi, ali KVM je i dalje prihvatljiv. Iz rezultata koji su dobiveni metodama prikazanim u radu vidljivo je da korisnici imaju razne mogućnosti kada je riječ o virtualiziranim rješenjima koji koriste grafičke kartice.

Ključne riječi: *hipervizor, virtualizacija, Hyper-V, KVM, grafička kartica, performanse, prosljeđivanje*

ABSTRACT

This paper explores and compares the capabilities of GPU passthrough on Hyper-V and KVM hypervisors to virtual machines, detailing the methods, procedures, and architectural differences that affect their implementation. A key challenge was ensuring testing was conducted on identical hardware, which provided reliable and comparable performance results across different operating systems. The goal of this paper is to offer a platform for testing the performance and compatibility of graphics cards with virtualized applications on different hypervisors. The aim was also to eliminate the need for specialized and expensive server systems and to enable the testing of graphics cards on commercially available hardware. As part of the practical section, system implementation steps were included, and the performance of the available graphics card was tested in all described scenarios. This created a practical environment on a single physical computer that allows for further research and replication. The results show that Hyper-V has a slight performance advantage, but KVM remains a viable option. From the results obtained using the methods presented in the paper, it is evident that users have various options when it comes to virtualized solutions that use graphics cards.

Keywords: *Hypervisor, Virtualization, Hyper-V, Graphic Cards, Performance, Passthrough*

1. UVOD
1. INTRODUCTION

U današnjim virtualiziranim okruženjima, kao što su računalstvo u oblaku i virtualizirani podatkovni centri, raste potreba za razinom performansi unutar virtualnih računala koje su donedavno bile rezervirane isključivo za fizička računala. Osnovni resursi kao što su jezgre procesora, radna memorija i diskovni prostor lako se dodjeljuju, dok je druge resurse teže dodijeliti.

Resursi koji su najpotrebniji s razvojem umjetne inteligencije i računalnog učenja su grafičke kartice. Njih nije moguće dodijeliti na isti način kao jezgre procesora već su za to implementirane različite metode prosljeđivanja ovisno o arhitekturi virtualizacijskog okruženja. [1] Grafičke kartice nisu jedina vrsta komponente koja se može proslijediti, ali su uz mrežne kartice je jedna od najvažnijih komponenti.

Prosljeđivanjem grafičkih kartica, virtualna računala dobivaju mogućnost identičnog pristupa hardveru kao i fizičko računalo. [2] Ovakav pristup omogućava veću iskoristivost sklopovlja te olakšava razvoj aplikacija kojima su potrebne grafičke kartice za izračune ili slične zadatke.

Ovaj rad prikazuje usporedbu svih dostupnih metoda prosljeđivanja grafičkih kartica na Hyper-V i KVM poslužiteljima. Rezultati testiranja svake od metoda uspoređuju se s osnovnom vrijednošću performansi dobivenom izravno na fizičkom hardveru. Metode za postizanje prosljeđivanja detaljno su opisane u trećem poglavlju, a za Hyper-V uključuju Hyper-V dodjelu kartice (DDA) i Hyper-V particioniranje (GPUP), dok se za KVM koristi isključivo fizička dodjela kartice (VFIO).

2. VIRTUALIZACIJA I HIPERVIZORI
2. VIRTUALIZATION AND HYPERVISORS

Virtualizacija se može definirati kao pokretanje nekoliko virtualnih računala na jednom fizičkom sklopovlju. Kroz računalnu povijest, virtualizacija i virtualizacijska rješenja dobivala su na važnosti i popularnosti zbog velikog broja prednosti koje donose. Jedan od glavnih pojmova kada se radi o virtualizaciji je hipervizor ili VMM (Virtual machine manager) [3].

Hipervizor je ključna softverska komponenta koja omogućuje stvaranje i upravljanje virtualnim strojevima, osiguravajući njihovu izolaciju od fizičkog hardvera i njihovu međusobnu izolaciju. Postoje dva tipa hipervizora koji se razlikuju po svojoj arhitekturi. Riječ je o hipervizorima prvog tipa te hipervizorima drugog tipa. Hipervizor prvog tipa pokreće se izravno na sklopovlju bez dodatnog sloja između hipervizora i same jezgre operativnog sustava. Takvi hipervizori su Hyper-V te KVM. Na taj način se postižu bolje performanse virtualnih računala jer se smanjuje potreba za prevođenjem poziva između virtualnog računala i samog sklopovlja [4]. Hipervizor drugog tipa se ne pokreće izravno na sklopovlju već su aplikacije na nekom operativnom sustavu. Takvi hipervizori su Oracle Virtualbox i VMware Player. Oni postižu lošije performanse te su ograničeni mogućnostima, ali virtualizaciju čine pristupačnima jer je takve hipervizore jednostavno instalirati i koristiti [4].

U sklopu ovog rada fokus je na hipervizore prvog tipa jer omogućavaju prosljeđivanje grafičkih kartica te optimizirane performanse u odnosu na hipervizore drugog tipa.

Hyper-V je Microsoftovo rješenje hipervizora. Hyper-V se nakon instalacije nalazi između samog operativnog sustava i komponenti računala te se virtualna računala dijele na osnovnu ili korijensku particiju koja jedina ima pristup sklopovlju te gostujućih particija koje iz nje nastaju. Na ovaj način particije tvore roditelj-dijete hijerarhijski odnos [5]. KVM (Kernel-based virtual machine) je implementacija za stvaranje virtualnih računala koja je uključena u samoj jezgri operativnog sustava Linux. KVM je postao dio Linux jezgre od njene inačice 2.6.20, što znači da je izravno integriran u jezgru operativnog sustava Linux. Zbog te bliske povezanosti s jezgrom, a time i sa samim sklopovljem, KVM se smatra hipervizorom prvog tipa, koji radi izravno na fizičkom računalu [6].

3. METODE PROSLJEĐIVANJA GRAFIČKIH KARTICA
3. GRAPHICS CARD FORWARDING METHODS

Svaki hipervizor ima različite metode prosljeđivanja grafičkih kartica. Tako Hyper-V

omogućava dva načina prosljeđivanja dok KVM omogućava jedan način.

Hyper-V DDA (Discreet Device Assignment) jedan je od načina prosljeđivanja grafičke kartice, ali i određenih drugih komponenti koje koriste PCI sabirnicu. Iako je Microsoftova preporuka koristiti skuplje grafičke kartice namijenjene podatkovnim centrima, moguće je koristiti i naizgled nepodržane kartice što smanjuje troškove implementacije. Budući da Microsoft ne podržava prosljeđivanje takvih kartica, u ovom radu pokazat će se kako se to može postići, što je korisno za uspostavu testnog okruženja na povoljnom hardveru. Ovakav pristup je pogodan za scenarije poput pisanja članka ili za kućnu upotrebu, iako je moguće da sve komponente grafičke kartice neće biti u potpunosti iskorištene jer ih Microsoft nije uvrstio na popis kompatibilnog hardvera.

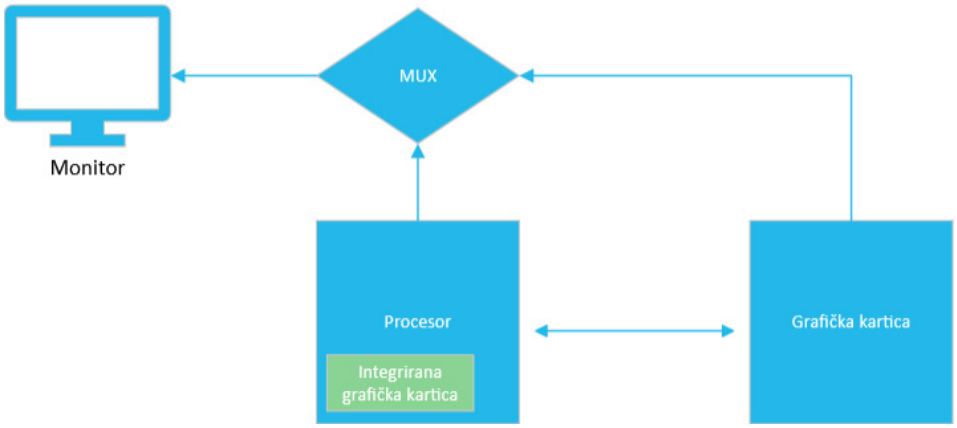
Kod DDA prosljeđuje se kartica isključivo jednom virtualnom računalu, bez mogućnosti dijeljenja s hipervizorom ili drugim virtualnim strojevima. Ova je metoda idealna kada virtualno računalo može u potpunosti iskoristiti sve mogućnosti kartice, na primjer kod zadataka kao što je obrada videozapisa, jer u suprotnom performanse neće biti u potpunosti iskorištene [7]. DDA se implementira putem svega nekoliko Powershell naredbi, što ga čini jednostavnijim za postavljanje od KVM-a i particioniranja.

Hyper-V particioniranje druga je metoda kojom se jedna fizička grafička kartica može podijeliti između hipervizora i više virtualnih računala. Umjesto cijele kartice, virtualnom se stroju prosljeđuje samo virtualni adapter, a hipervizor upravlja komunikacijom s fizičkim hardverom. Ova metoda korisna je kada virtualnim strojevima grafičke kartice nisu potrebne u punom kapacitetu, čime se maksimizira iskorištenost hardvera i smanjuju troškovi. Nedostatak particioniranja je što se može primijeniti samo na grafičke kartice, za razliku od DDA-a koji podržava i druge komponente. Dodatno je važno naglasiti da se pri podjeli grafičke kartice performanse smanjuju ovisno o tome koliko virtualnih računala jednu grafičku karticu koristi [6]. Sama implementacija je nešto zahtjevnija od DDA pristupa jer je potrebno ručno kopirati datoteke upravljačkog programa s hipervizora na virtualno računalo. DDA za razliku od toga podržava normalnu instalaciju upravljačkog programa bez potrebe za ručnim postavljanjem.

KVM prosljeđivanje metoda je dostupna na operativnom sustavu Linux, a po ponašanju je slična rješenju Hyper-V DDA. Proslijeđena grafička kartica je dostupna samo virtualnom računalu te se ne može zajednički dijeliti. Da bi funkcioniralo, grafička kartica na hipervizoru mora koristiti upravljački program VFIO (Virtual function I/O) kroz koji KVM osigurava izolaciju komponente od operativnog sustava. Glavna prednost KVM-a je što je cjenovno prihvatljivije rješenje jer ne zahtijeva licenciranje niti posebne, preporučene grafičke kartice. Ipak, postavljanje je zahtjevnije i traži dublje razumijevanje virtualizacijskih tehnologija uz isti nedostatak kao kod DDA – grafička je kartica nedostupna hipervizoru i drugim virtualnim računalima [8].

4. OKOLINA I ALATI TESTIRANJA
4. TESTING ENVIRONMENT AND TOOLS

Testno okruženje postavljeno je na prijenosnom računalu koje, da bi se reproduciralo, zahtijeva poseban preduvjet – postojanje dviju grafičke kartice i komponente zvane multipleksor (MUX). Kod većine prijenosnih računala najčešće postoji samo jedan videoizlaz povezan s integriranom grafičkom karticom procesora, dok diskretna grafička kartica šalje sliku integriranoj kartici na prikaz. Multipleksor omogućuje da grafičke kartice budu odvojene i da šalju signale na odvojene ulaze multipleksora koje on šalje na zajednički izlaz ovisno o izboru aktivne grafičke kartice. Takvo je odvajanje ključno kako bi hipervizor mogao koristiti isključivo onu grafičku karticu koja se neće prosljeđivati, čime se sprječava gubitak video signala na samom hipervizoru. Multipleksor nije potreban na stolnim računalima jer grafičke kartice uglavnom imaju vlastite video izlaze. Slika 1 prikazuje arhitekturu korištenog prijenosnog računala s ugrađenim multipleksorom.



Slika 1 *Arhitektura prijenosnog računala s multipleksorom*

Figure 1 *Laptop architecture with included multiplexer*

Konfiguracija testnog prijenosnog računala sastoji se od sljedećih komponenti koje su prikazane u Tablici 1.

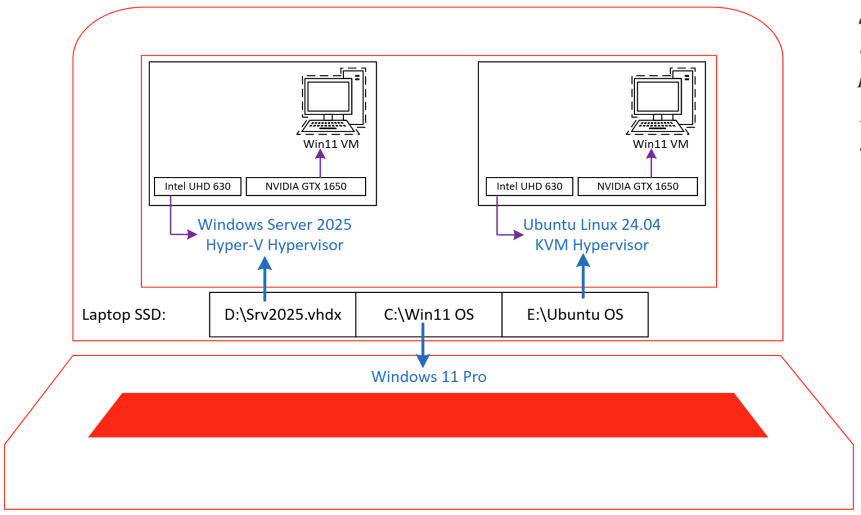
Tablica 1. *Hardverske komponente testnog računala*

Table 1. *Test computer hardware components*

Vrsta komponente	Vrijednost
Procesor	Intel i5 9300h 4 jezgre, 4 virtualne jezgre
Grafička kartica 1	Intel UHD 630
Grafička kartica 2	NVIDIA GTX 1650
Radna memorija	16 GB DDR4
Disk 1	Samsung 970 EVO NVMe 512 GB
Disk 2	Western Digital 1TB HDD
Operativni sustav	Windows 11 Pro

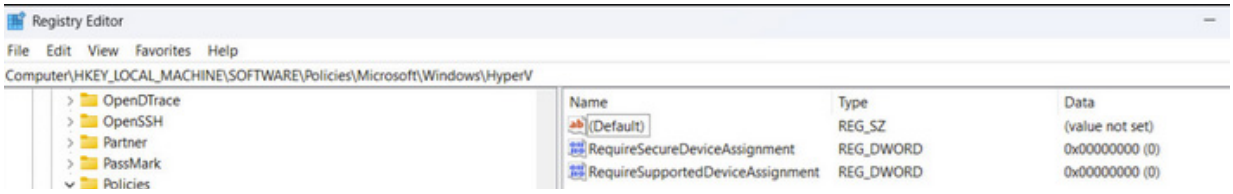
Budući da je cilj rada bio osigurati rad različitih hipervizora koji se izvršavaju nad različitim operativnim sustavima koristeći jedno fizičko računalo, konačno je rješenje kompleksni set instaliranih operativnih sustava, virtualnih strojeva, kao i sam redoslijed njihove instalacije. Kao što je ranije spomenuto, razlog za ovaj pristup je mogućnost paralelnog mjerenja performansi na identičnom hardveru, čime se smanjuju konačni troškovi i dobivaju pouzdani rezultati bez varijabli uzrokovanih hardverskim razlikama. Ukupni prikaz cjelokupnog virtualnog okruženja postavljenog na prijenosom računalu prikazan je Slikom 2.

Nakon instalacije osnovnog operativnog sustava Windows 11, postavljeno je testno okruženje za Hyper-V. To je učinjeno izradom virtualnog diska u VHDX formatu na zasebnoj particiji te instaliranjem operativnog sustava Windows Server 2025 na taj virtualni disk. Nakon toga virtualni disk u boot manager dodan je popis operativnih sustava, čime je omogućeno pokretanje Windows Servera s istog fizičkog računala na kojem se nalazi Windows



Slika 2 *Shematski prikaz lokacije i instalacije operativnih sustava na prijenosnom računalu*

Figure 2 *Schematic diagram of OS location and installation on a laptop*



Slika 3 *Modifikacija zapisa Windows Registry radi podrške za grafičku karticu na testnom laptopu*

Figure 3 *Modifying Windows Registry entries to support the graphics card on the test laptop*

```
PS C:\Users\Administrator> Set-VM -Name “Windows 10” -AutomaticStopAction TurnOff
PS C:\Users\Administrator> Set-VM -GuestControlledCacheTypes $true -VMName “Windows 10”
PS C:\Users\Administrator> Set-VM -LowMemoryMappedIoSpace 3Gb -VMName “Windows 10"
PS C:\Users\Administrator> Set-VM -HighMemoryMappedIoSpace 33280Mb -VMName “Windows 10”
PS C:\Users\Administrator>
Dismount-VMHostAssignableDevice -Force -LocationPath "PCIROOT(0)#PCI(0100)#PCI(0000)"
> Add-VMAssignableDevice -LocationPath "PCIROOT(0)#PCI(0100)#PCI(0000)" -VMName “Windows 10”
```

Slika 4 *Naredbe u PowerShellu za pripremu virtualnog računala kod DDA implementacije*

Figure 4 *PowerShell commands for preparing a virtual machine for DDA deployment*

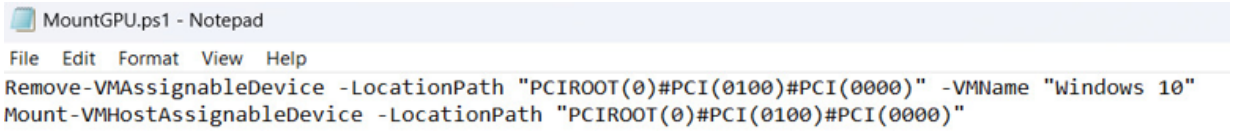
11. Unutar Windows Servera, instaliran je Hyper-V pomoću kojega je stvoreno virtualno računalo pod nazivom "Windows 10" s dodijeljenim resursima (4 jezgre procesora, 8 GB radne memorije i 60 GB diskovnog prostora). Na ovom virtualnom računalu provedena su testiranja DDA i Hyper-V particioniranja.

Sljedeće je za potrebe KVM testiranja na SSD disku stvorena nova particija na koju je instaliran Ubuntu Linux. KVM je odabran jer je dio jezgre operativnog sustava Linux, a za upravljanje virtualnim računalima instaliran je popratni alat virt-manager. Virtualno računalo stvoreno je s istom konfiguracijom kao i Hyper-V virtualno računalo (4 jezgre procesora, 8 GB radne memorije i 60 GB diskovnog prostora) kako bi virtualni strojevi s kojih se provodi testiranje bili identični.

Za DDA implementaciju koristile su se gotove skripte Microsoft PowerShell za detekciju kompatibilnih grafičkih kartica. Budući da testna grafička kartica na prijenosnom računalu ne udovoljava Microsoftovim preporukama, ručno je premošteno ovo ograničenje unosom postavki u Windows registar. Ovaj postupak nije službeno dokumentiran od strane Microsofta, ali je bio ključan za omogućavanje testiranja na nepodržanom hardveru, što je korisno za buduća istraživanja ili kućnu upotrebu. Važno je napomenuti da zbog ručnog premošćivanja, performanse kartice mogu biti smanjene, jer nisu sve njezine komponente u potpunosti iskorištene. Vrijednosti koje je potrebno postaviti prikazane su na Slici 3.

Nakon ovog koraka za aktivaciju DDA prosljeđivanja mogu se pratiti službene upute s Microsoftovih mrežnih stranica koje se sastoje od deaktivacije grafičke kartice na fizičkom računalu, spajanja kartice na virtualno računalo putem naredbi u PowerShellu te instalacije upravljačkih programa u virtualno računalo za proslijeđenu grafičku karticu. Na Slici 4 prikazani su parametri naredbi u PowerShell kako se osigurava ispravno gašenje računala kao i dostupna memorija za grafičku karticu, dok posljednje dvije naredbe služe za spajanje kartice na virtualno računalo [7].

Za Hyper-V particioniranje potrebno je vratiti grafičku karticu hipervizoru te ju ponovno uključiti u Upravitelju uređaja. Potrebne naredbe vidljive su u Slici 5.



Slika 5 *PowerShell skripta za vraćanje grafičke kartice hipervizoru*

Figure 5 *PowerShell script to restore graphics card to the hypervisor*

```
GPUP.ps1 - Notepad
File  Edit  Format  View  Help
$vm = "Windows 10"
if (Get-VMGpuPartitionAdapter -VMName $vm -ErrorAction SilentlyContinue) {
   Remove-VMGpuPartitionAdapter -VMName $vm
}
Set-VM -GuestControlledCacheTypes $true -VMName $vm
Set-VM -LowMemoryMappedIoSpace 1Gb -VMName $vm
Set-VM -HighMemoryMappedIoSpace 32Gb -VMName $vm
Add-VMGpuPartitionAdapter -VMName $vm
```

Slika 6 *Naredbe potrebne za particioniranje grafičke kartice u skripti*

Figure 6 *Required commands for graphics card partitioning*

```
#!/bin/bash
for d in /sys/kernel/iommu_groups/*/devices/*; do
  n=${d#*/iommu_groups/*}; n=${n%%/*}
  printf 'IOMMU Group %s ' "$n"
  lspci -nns "${d##*/}"
done
```

Slika 7 *Skripta za ispis IOMMU adresa*

Figure 7 *Script for printing IOMMU addresses*

Dodatno se za pripremu trebaju kopirati datoteke upravljačkog programa s hipervizora u virtualno računalo. Obična instalacija nije dovoljna jer upravljački programi moraju biti identični [6]. Nakon priprema mogu se pokrenuti naredbe koje se zbog jednostavnosti mogu staviti u skriptu. Skripta priprema virtualno računalo slično kao kod DDA rješenja te dodaje adapter particije u virtualno računalo.

Za konfiguraciju prosljeđivanja grafičke kartice na KVM Linux hipervizoru potrebno je osigurati da operativni sustav ne koristi grafičku karticu koja se prosljeđuje. To se može postići uređivanjem samih postavki sustava, ali se mogu koristiti i posebni alati. Za te potrebe u ovom radu koristit će se supergfxctl. Radi se o alatu koji omogućava promjenu grafičkih kartica koje operativni sustav koristi, ali dodatno omogućava uključivanje VFIO upravljačkog programa za grafičku karticu koji je preduvjet za prosljeđivanje.

Prije samog prosljeđivanja potrebno je osigurati da je IOMMU (Input/Output Memory management unit) korišten. IOMMU pruža virtualne adrese preko kojih se prosljeđivanje odrađuje. Da bi se IOMMU aktivirao na sklopovlju ovog prijenosnog računala potrebno je u postavke bootloadera na putanji /etc/default/grub dodati intel_iommu=on te ponovno pokrenuti računalo. Nakon što je IOMMU uključen, moguće je pomoću skripte otkriti adresu grafičke kartice koja će se koristiti za prosljeđivanje [10].

Za prosljeđivanje potrebno je u XML konfiguracijskoj datoteci definirati uređaj te njegove adrese, ali i u sučelju dodati sami PCI-e uređaj.

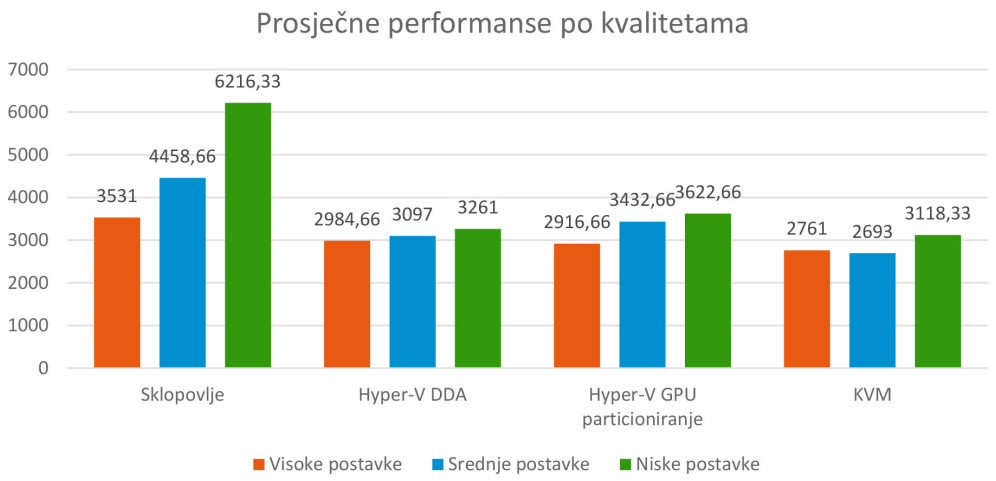
Nakon prosljeđivanja i instalacije upravljačkog programa grafička kartica je dostupna virtualnom računalu i spremna za testiranje. Za testiranje performansi koristi se Unigine Heaven 4.0 Benchmark alat, koji testira sposobnosti grafičkih kartica iscrtavanjem grafički zahtjevnih scena dok rezultate prikazuje u obliku bodova dobivenih izračunom prosječnog, minimalnog i maksimalnog broj sličica u sekundi (Frame per Second) za svaku scenu. Alat omogućava detaljno namještanje različitih grafičkih postavki, uključujući kvalitetu tekstura, sjene, anti-aliasing (AA), globalno osvjetljenje.

Radi jednostavnosti replikacije i usporedivosti rezultata, umjesto ručnog postavljanja opcija, za testiranje su odabrani zadani predlošci unutar alata: niske, srednje i visoke grafičke postavke. Sva su testiranja provedena na rezoluciji 1280x800

```
</devices>
<qemu:override>
  <qemu:device alias="hostdev0">
    <qemu:frontend>
      <qemu:property name="x-pci-sub-vendor-id" type="unsigned" value="6058"/>
      <qemu:property name="x-pci-sub-device-id" type="unsigned" value="16379"/>
    </qemu:frontend>
  </qemu:device>
</qemu:override>
```

Slika 8 *XML konfiguracija virtualnog računala s adresama uređaja u dekadskom formatu*

Figure 8 *XML configuration of the virtual computer with device addresses in decimal format*



Slika 9 *Rezultati sklopovlja i testiranih implementacija prosljeđivanja*

Figure 9 *Results of hardware and tested forwarding implementations*

zbog ograničenja virtualnih računala. Uzrok tog ograničenja je što je u virtualnim računalima zadana virtualna grafička kartica ostaje primarna, dok se proslijeđena koristi odvojeno. Iako takav pristup uvelike olakšava konfiguraciju, kao posljedicu ima ograničenu rezoluciju. Spomenute postavke utječu na grafičku kvalitetu samih scena; što su postavke testa više, to su scene grafički zahtjevnije, što rezultira padom broja bodova [11].

5. REZULTATI
5. RESULTS

Za testiranje korišten je ranije spomenuti alat u verziji DirectX 9. Budući da testovi pokazuju zanemariva odstupanja pri svakom pokretanju, svaki je test ponovljen svega tri puta kako bi se dobila pouzdana prosječna vrijednost i izbjegla potreba za većim brojem mjerenja. Veći broj ponavljanja imao bi smisla samo u slučaju da se rezultati uspoređuju s drugim istraživanjima provedenim na identičnom hardveru, što bi osiguralo veću preciznost. Međutim, činjenica da će se buduća testiranja vjerojatno razlikovati u hardverskim postavkama uklanja potrebu za većim brojem ponavljanja u svrhu dobivanja preciznijih rezultata.

Dobiveni rezultati pokazuju slične vrijednosti u odnosu na mogućnosti sklopovlja pri pokretanju testa u visokim postavkama. Pri smanjivanju postavki kvalitete, rast rezultata nije linearan kao na samom fizičkom sklopovlju. Kako se kvaliteta smanjuje opterećenje grafičke kartice opada, dok opterećenje procesora raste. Ovaj je fenomen poznat kao CPU/GPU usko grlo (engl. bottleneck), gdje se pri visokim postavkama GPU smatra uskim grlom, dok pri nižim postavkama procesor postaje usko grlo jer mora brže pripremati podatke za grafičku karticu [12].

Dodatno, virtualna računala nemaju dodijeljen jednak broj virtualnih jezgri procesora kao fizičko računalo, a sam hipervizor izdvaja dio dodijeljene virtualne procesorske snage za uspostavu i održavanje izolaciju virtualnog računala, što smanjuje ukupne dostupne performanse.

Ostale razlike u rezultatima odražavaju razlike među implementacijama. Ovakvi rezultati ukazuju na dobru iskoristivost sklopovlja unutar samih virtualnih računala kada su zadaci uvelike ovisni o grafičkim karticama, ali ne i o procesoru. Važno je napomenuti da KVM ima malo lošije, ali ipak jako dobre rezultate u odnosu na Hyper-V. To je jako dobro s obzirom na to da KVM ne zahtijeva licence te je iz tog razloga pristupačniji. Dobiveni rezultati usporedivi su s omjerima učinkovitosti prosljeđivanja grafičkih kartica dobivenih u drugim istraživanjima u ovom području. To potvrđuje da virtualizacija GPU-a u sličnim uvjetima postiže performanse vrlo bliske onima na fizičkom hardveru, čime se dodatno opravdava njihova primjena u poslovnim i istraživačkim okruženjima [2] [13] [14].

6. ZAKLJUČAK
6. CONCLUSION

Usporedba metoda prosljeđivanja grafičkih kartica u Hyper-V i KVM okruženju pokazuje da svaki hipervizor ima svoje prednosti. Hyper-V je jednostavniji za postavljanje te bolje

prilagođen za Microsoftov ekosustav, ali zahtijeva specifično, često skuplje sklopovlje za postizanje prosljeđivanja. Hyper-V je zbog zatvorenijeg sustava jednostavnije postaviti te pruža bolje performanse bez dodatnih postavki. Iz tog razloga se može reći da je Hyper-V dobro rješenje za implementacije koje zahtijevaju najbolje moguće performanse, a da samo virtualno okruženje nije komplicirano za postaviti. S druge strane, KVM je fleksibilniji, besplatan i podržava različito sklopovlje, ali je samo postavljanje kompliciranije i zahtjeva više znanja. KVM također ne postiže jednako dobre rezultate bez dodatnih postavki. S obzirom na ove razlike KVM bi bio idealan za korisnike koji uglavnom koriste Linux u svojim okruženjima, posjeduju jako dobro znanje o operativnim sustavima i virtualizaciji te ne žele plaćati dodatne licence ili posebne grafičke kartice.

Performanse kod grafički zahtjevnih zadataka su slične te izbor uglavnom ovisi o operativnom sustavu hipervizora koje se želi koristiti. Oba rješenja nude i grafička sučelja koja olakšavaju upravljanje hipervizorom i virtualnim računalima.

S rastućim zahtjevima za virtualna računala zbog zadataka kao što su umjetna inteligencija i računalno učenje, prosljeđivanje grafičkih kartica postaje ključna tehnologija koja omogućuje snažne performanse unutar virtualnih okruženja, uz istovremeno smanjenje troškova razvoja i testiranja. Virtualizacija tako dobiva sve širu primjenu – od malih ureda do podatkovnih centara i oblaka jer omogućava lakše održavanje, upravljanje i manji gubitak performansi.

U budućnosti bi se ove tehnologije mogle razviti još više u smjeru jednostavnosti kako bi se što bolje iskoristile sposobnosti modernih grafičkih kartica ne samo u podatkovnim centrima već i u kućanstvu pokretanjem nekoliko operativnih sustava istovremeno za više korisnika i slične zadatke.

7. REFERENCE

7. REFERENCES

[1.] Dowty, Micah & Sugerman, Jeremy. (2008). GPU virtualization on VMware's hosted I/O architecture. Operating Systems Review - SIGOPS. 43. doi: 10.1145/1618525.1618534.

[2.] J. P. Walters et al., "GPU Passthrough Performance: A Comparison of KVM, Xen, VMWare ESXi, and LXC for CUDA and OpenCL Applications," 2014 IEEE 7th International Conference on Cloud Computing, Anchorage, AK, USA, 2014, pp. 636-643, doi: 10.1109/CLOUD.2014.90.

[3.] M. Portnoy, Virtualization Essentials. John Wiley & Sons, 2012.

[4.] A. Desai, R. Oza, P. Sharma, i B. Patel, „Hypervisor: A Survey on Concepts and Taxonomy“, sv. 2, izd. 3, 2013.

[5.] scooley, „Hyper-v Architecture“. Pristupljeno: 01. lipanj 2025. [Na internetu]. Dostupno na: https://learn.microsoft.com/en-us/virtualization/hyper-v-on-windows/reference/hyper-v-architecture

[6.] „What is KVM?“ Pristupljeno: 01. lipanj 2025. [Na internetu]. Dostupno na: https://www.redhat.com/en/topics/virtualization/what-is-KVM

[7.] meaghanlewis, „Deploy graphics devices by using Discrete Device Assignment“. Pristupljeno: 01. lipanj 2025. [Na internetu]. Dostupno na: https://learn.microsoft.com/en-us/windows-server/virtualization/hyper-v/deploy/deploying-graphics-devices-using-dda

[8.] meaghanlewis, „Partition and share GPUs with virtual machines on Hyper-V“. Pristupljeno: 01. lipanj 2025. [Na internetu]. Dostupno na: https://learn.microsoft.com/en-us/windows-server/virtualization/hyper-v/gpu-partitioning

[9.] „Chapter 1. GPU device passthrough: Assigning a host GPU to a single virtual machine | Setting up an NVIDIA GPU for a virtual machine in Red Hat Virtualization | Red Hat Virtualization | 4.3 | Red Hat Documentation“. Pristupljeno: 01. lipanj 2025. [Na internetu]. Dostupno na: https://docs.redhat.com/en/documentation/red_hat_virtualization/4.3/html/setting_up_an_nvidia_gpu_for_a_virtual_machine_in_red_hat_virtualization/proc_nvidia_gpu_passthrough_nvidia_gpu_passthrough

[10.] 262588213843476, „List the devices and their IOMMU groups.“, Gist. Pristupljeno: 24. lipanj 2025. [Na internetu]. Dostupno na: https://gist.github.com/flungo/428c374c040de1d0a30fd4a593d39040

[11.] UNIGINE HEAVEN BENCHMARK 4.1 Pristupljeno 14. srpanj 2025. [Na internetu]. Dostupno na: https://benchmark.unigine.com/docs/heaven/User_Manual.pdf

[12.] "What Is a CPU Bottlenecking & How to Fix It," Lenovo.com. Pristupljeno 25. lipanj 2025. [Na internetu]. Dostupno na : https://www.lenovo.com/us/en/glossary/cpu-bottlenecking/.

[13.] A. J. Younge, J. P. Walters, S. Crago and G. C. Fox, "Evaluating GPU Passthrough in Xen for High Performance Cloud Computing," 2014 IEEE International Parallel & Distributed Processing Symposium Workshops, Phoenix, AZ, USA, 2014, pp. 852-859, doi: 10.1109/IPDPSW.2014.97.

[14.] R. Radhakrishnan, Y. Varma and U. Kurkure, "Evaluating GPU Performance for Deep Learning Workloads in Virtualized Environment," 2019 International Conference on High Performance Computing & Simulation (HPCS), Dublin, Ireland, 2019, pp. 904-908, doi: 10.1109/HPCS48598.2019.9188098.

AUTORI · *AUTHORS*

• **Nikša Kovačić** - student je druge godine diplomskog studija Računalne mreže i sustavi na Tehničkom veleučilištu u Zagrebu. Uz studij također je zaposlen kao administrator sustava gdje koristi znanje stečeno kako na studiju tako i privatnim radom. U slobodno vrijeme koristi vlastitu instalaciju Proxmox za podizanje virtualnih poslužitelja u svrhu učenja tehnologija kao što su Docker i Kubernetes.

Korespondencija · *Correspondence*

nkovacic2@tvz.hr

• **Ognjen Mitrović** - Nepromijenjena biografija nalazi se u časopisu Polytechnic & Design Vol. 5, No. 2, 2017.

Korespondencija · *Correspondence*

ognjen.mitrovic@tvz.hr

• **Vedran Tadić** - Nepromijenjena biografija nalazi se u časopisu Polytechnic & Design Vol. 11, No. 2, 2023.

Korespondencija · *Correspondence*

vedran.tadic@tvz.hr

• **Tin Antonić** - rođen je 1993. godine u Zagrebu. Završio je X. gimaziju Ivan Supek u Zagrebu. Stručni i diplomski studij završio je na Tehničkom veleučilištu u Zagrebu na kojem je diplomirao 2019. godine. Tema diplomskog rada kojeg je obranio je “Odabir skupa boja u dizajnu grafičkih sučelja”. Zaposlen je u OŠ Novo Čiče od 2019. godine gdje predaje informatiku. Kao vanjski suradnik na TVZ-u izvodi laboratorijske vježbe od akademske godine 2020/2021 u zvanju asistenta. Laboratorijske vježbe drži na predmetu Programiranje na stručnom prijediplomskom studiju Elektotehnika.

Korespondencija · *Correspondence*

tantonic@tvz.hr