CYBERUS TECHNOLOGY

The KVM Backend for VirtualBox

Julian Stecklina, Martin Messer

KVM Forum 2024

Disclaimer 1

VirtualBox is a trademark owned by Oracle. This project has no affiliation with Oracle.

Disclaimer 2

This is teamwork with Thomas Prescher, Stefan Kober, Sebastian Eydam and many other colleagues at Cyberus. We are just the lucky ones to present. ☺

Outline

- Introduction: Cyberus Technology
- Why KVM for VirtualBox?
- Zoom into Nested Virtualization Issues
- Summary

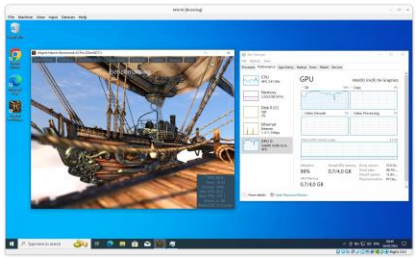
Cyberus Technology: An Overview

About us

- Founded 2017
- Self-funded, boot-strapped deep-tech firm
- Profitable, no debt, no VC
- HQ: **Dresden**, ~25 employees
- **Background:**

Claim to fame

- Meltdown & Spectre SC discovery
- Open-Source release of KVM backend for VirtualBox
- Long term virtualization partner for Germany's prime IT security provider
- Engagement for virtualization stack BSI approval



CYBERUS TECHNOLOGY

Open-Source Engagements

- Cloud Hypervisor / VirtualBox
- Linux KVM
- NixOS / nixpkgs

Hardware Test Automation

SoTest: On-HW synchronous Test Automation

- Enables agile SW product releases in virtualization technology development
- Short feedback cycles
- Discovery of functional and quality issues
- Automatic execution of all SW tests, on all OS's on all HW and platform variants on **every** code change

👉 See our other talk:
Automated Hypervisor Testing and Benchmarking

Hardware Orchestration and Testing

- Automated, remote, scalable
- Terminal Server
- Multi-Monitor Testing
- Creative Low-Level Solution

What is VirtualBox?

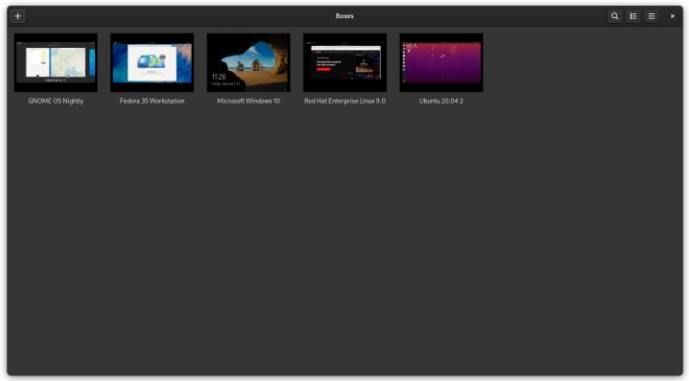
- A user-friendly virtualization software
  - Broad host and guest operating system support
  - No fiddling: Runs everything from DOS to Linux to Windows 11

- Powerful Guest Integration
  - Dynamic display resizing
  - Clipboard integration
  - Shared Folders
  - Drag’n’drop of files

- Great out-of-the-box experience

- **Lots of users!**

Relation to Qemu/Libvirt/Gnome Boxes/virt-manager…



libvirt VIRTUALIZATION API

QEMU

KVM

“Bazaar”

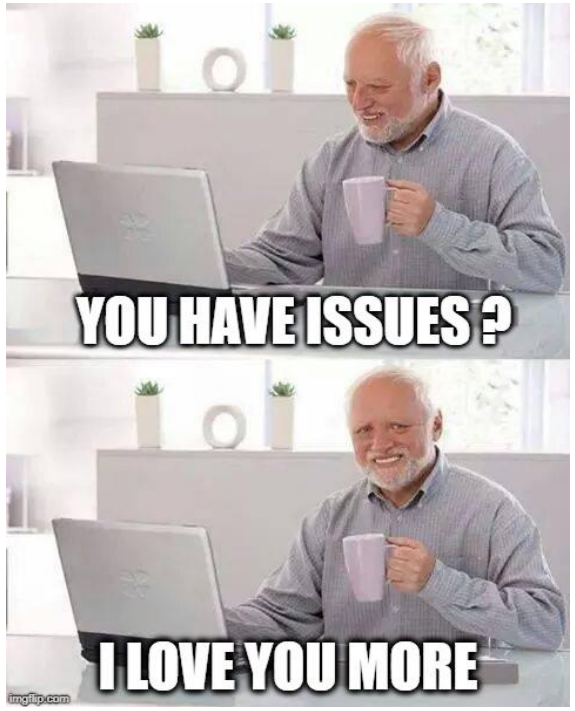
ORACLE VirtualBox

“Cathedral”

VirtualBox Issues

- VirtualBox ships their own “Type 2” (hosted) hypervisor
  - Linux kernel module `vboxdrv`
  - Historically had quality issues.
  - Takes care of lots of emulation tasks in the kernel. 👀
  - **Can’t use at the same time as KVM.**
- Third-party kernel modules are unpopular…
  - Need to use DKMS …
  - Users don’t like them.
  - OS vendors don’t like them.

-> Get rid of `vboxdrv`!



KVM Instead?

- Large relevant feature set!
  - APICv/AVIC, Hyper-V enlightenments, …
- Robust security
  - Vulnerabilities tend to be in less used / new parts
    - AMD SEV, Nested Virtualization, ...
    - Guest escapes using KVM bugs are rare!
- Works on all Linux versions
  - It's enabled on all Linux distros
  - Even with kernel hardening (grsecurity/PaX)
  - Basic feature set is stable for a long time

How do we marry VirtualBox and KVM?

VirtualBox Backends

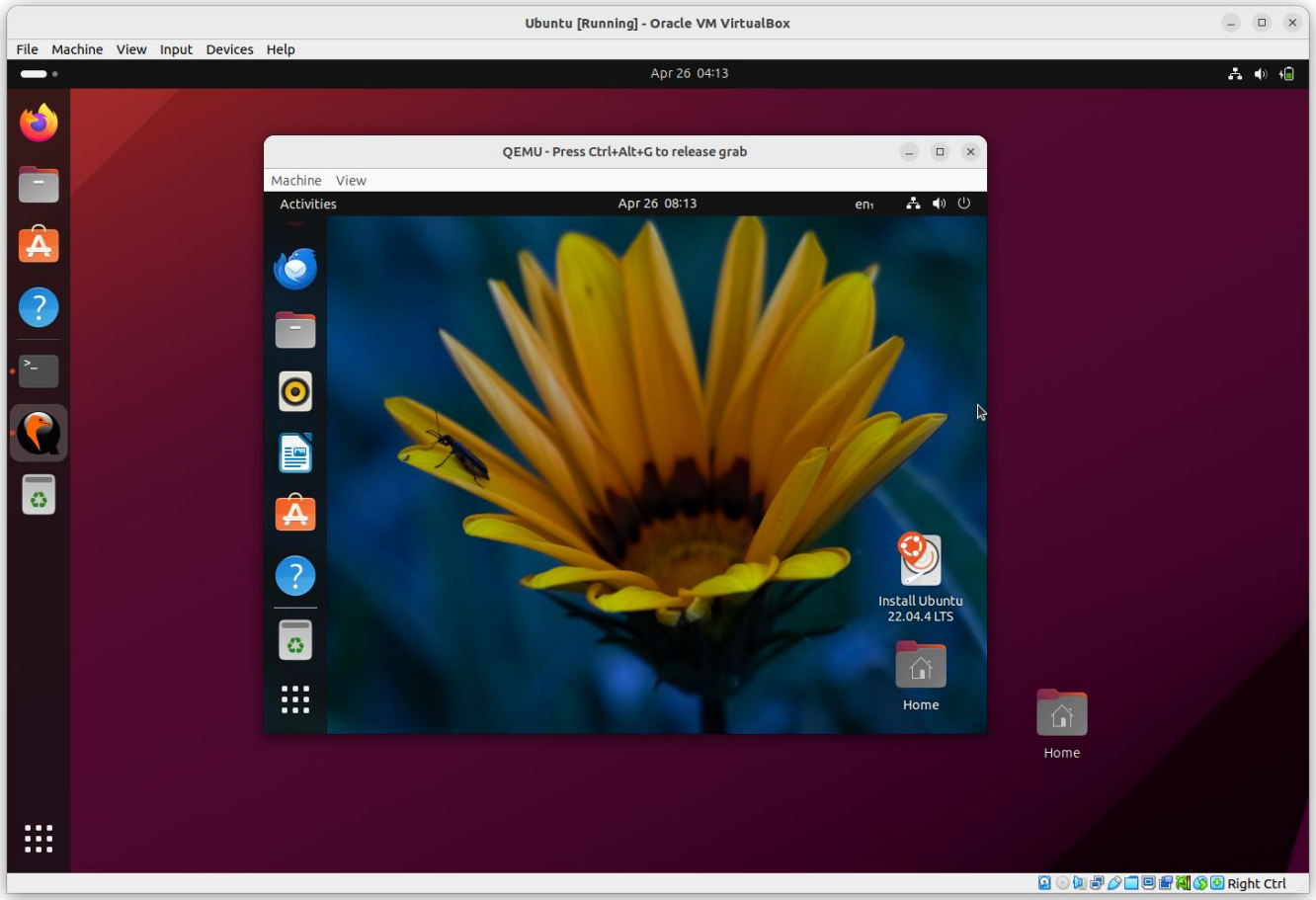
- VirtualBox has multiple backend types
  - HM – their own hypervisor (vboxdrv on Linux/Solaris/BSD/Windows)
  - NEM – “Native Execution Manager”, relies on default hypervisor of the platform
- VirtualBox includes proof-of-concept NEM backends for
  - Hyper-V
  - MacOS
  - **KVM**
- But disabled for Linux by default, because incomplete/broken…

Fixing the KVM Backend

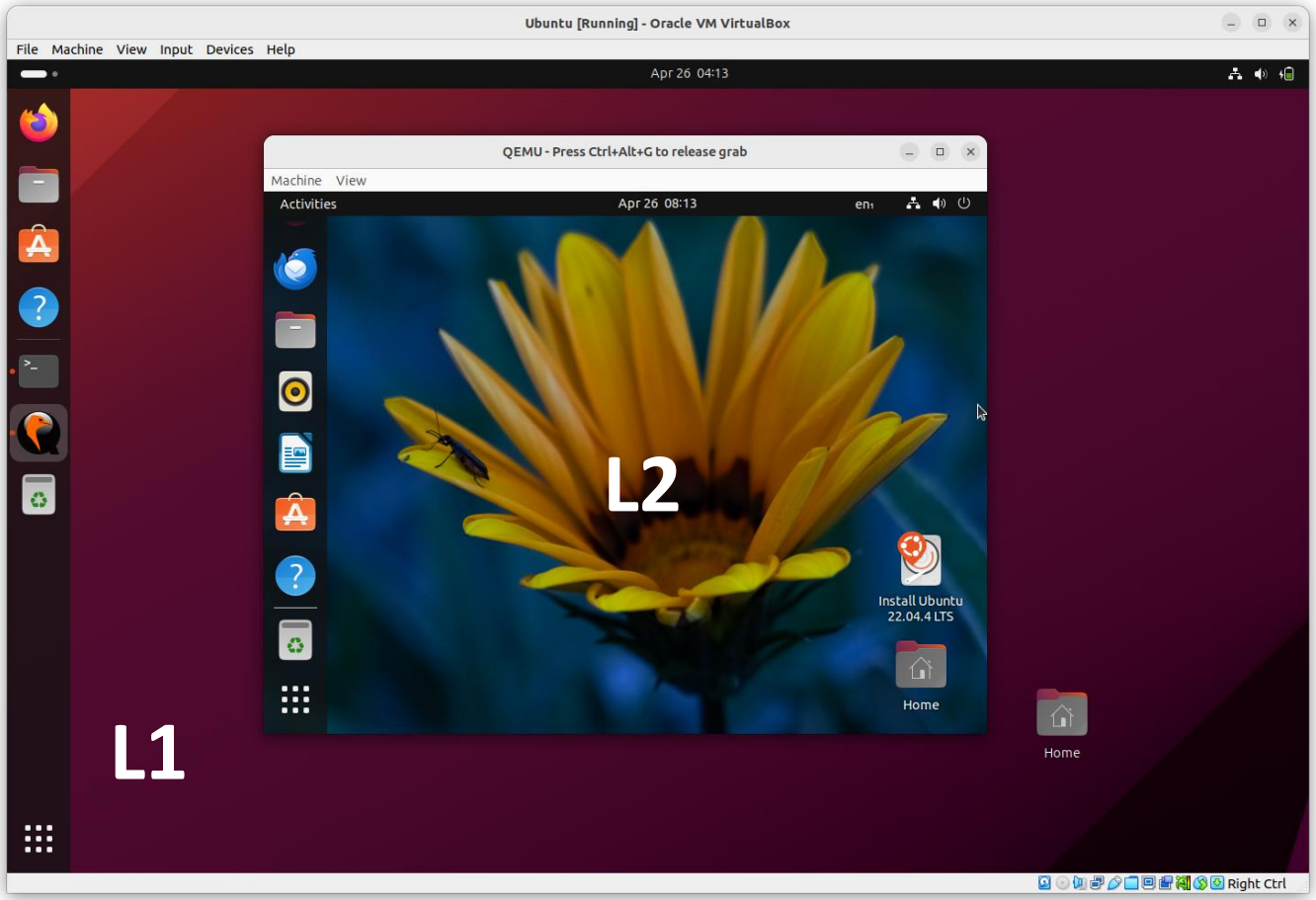
We invested the time to make it awesome!

- Making it work
  - Fixed Interrupt handling
  - Fixed Timeout handling
- Making it fast
  - Using KVM's Split-IRQCHIP feature
  - Using KVM's Hyper-V enlightenments
    - SYNIC / STIMER
- No time for all, our focus today: **Nested Virtualization Support**
  - Enable CPUID VMX bit, whitelist VMX MSRs and be happy?

Deep Dive: Nesting Virtualization with the KVM Backend



Quick Recap: Terminology

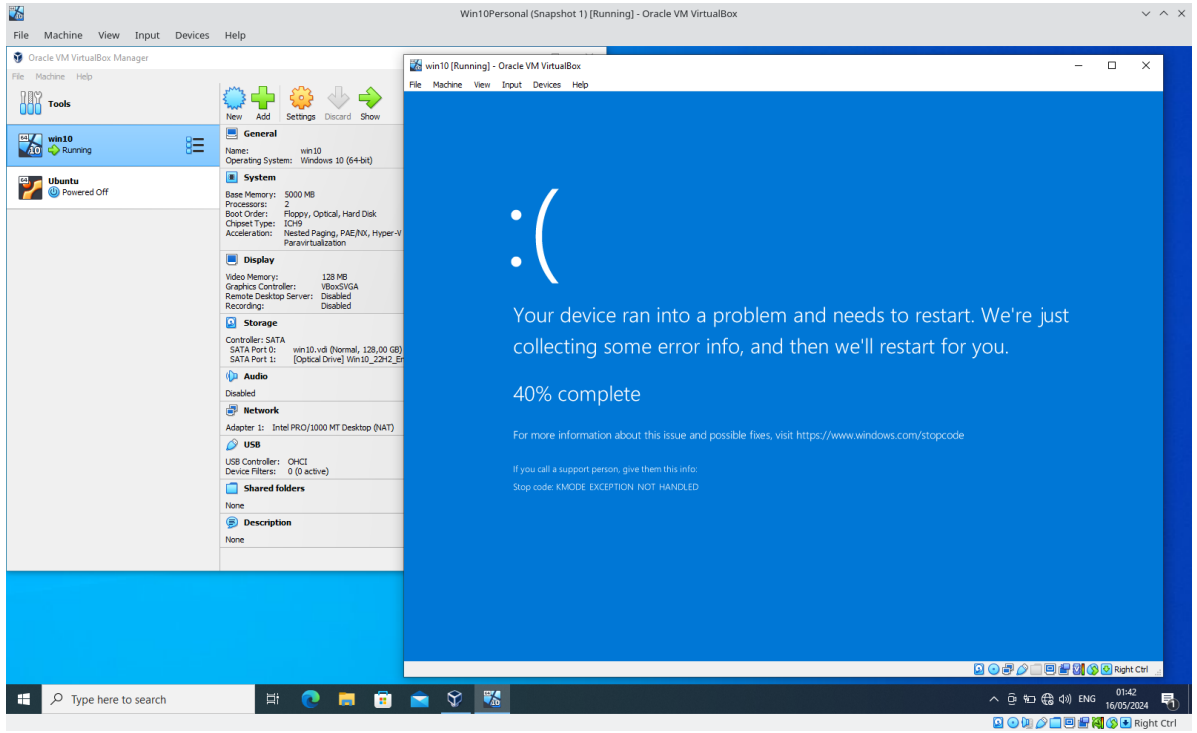


Mysterious Crashes in nested VMs?

Windows crashes during boot due to kernel exceptions.

Only occurred with:

- Nested VMs
- KVM backend for VirtualBox
- Windows as nested VM



Problematic code: TLB Flush for global pages

```
mov rax, cr4  ; Get CR4
mov rcx, rax  ; Remember the old value

btc rax, 7    ; Toggle CR4.PGE
mov cr4, rax  ; #GP! <- Shouldn't happen!

mov cr4, rcx  ; Restore old value
```

Guest Tests to the Rescue!

- We developed a set of low-level virtualization tests.
  - Open Source: https://github.com/cyberus-technology/guest-tests
  - Originally regression tests for our own microhypervisor.
  - Tests interrupt injection, VMX, …
  - Overlap with KVM Unit Tests.
- We reused nested virtualization tests to reproduce the issue.

Root Cause: CR4 leaks from L1 to L2

- When a signal causes an exit during L2 operation, the `struct kvm_run` contains the L2 state.
  - We use signals to interrupt vCPU for timer events.
- VBox always marks CR4 as dirty
  - KVM_SET_SREGS with unchanged CR4
  - Inefficient but no problem?
- KVM writes the L1 CR4 value to L2 CR4
  - Unexpected!
- Result: L1 CR4 leaks into L2
  - Guest sees CR4.VMXE
  - `MOV EAX, CR4; MOV CR4, EAX` -> #GP fault
- Similar issues in Qemu/KVM
  - `savevm` triggers [the issue](#) (qemu#2582)

```
test case: test_tinivisor_cr4
[INF main.cpp:176] CR4 is: 20
[INF main.cpp:176] CR4 is: a0
[INF main.cpp:176] CR4 is: 20
[INF main.cpp:176] CR4 is: a0
...
[INF main.cpp:176] CR4 is: 20
[INF main.cpp:176] CR4 is: a0
[INF main.cpp:176] CR4 is: 20
[INF main.cpp:183] Invalid cr4: 2020
Assertion failed @ test/guest/tinivisor/main.cpp:185
```

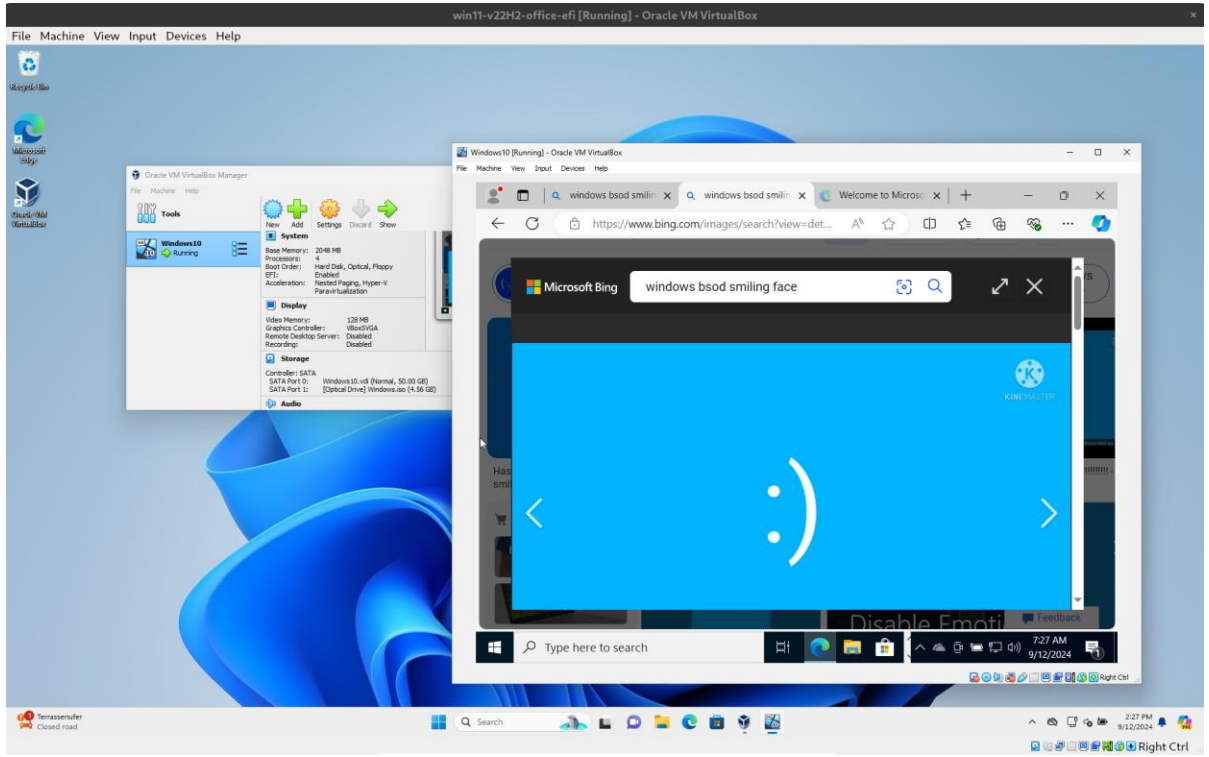
Workaround: Do what Qemu does

- Marking unchanged control registers as dirty in `struct kvm_run` causes the problem
- Qemu: Updates to Control Registers only on changed values
- We changed the behavior accordingly
  - But doesn’t seem like a robust solution!

Towards a Better Solution: API Changes

- Initial patch submission failed, but we got good discussions on the mailing list
- Proposed change in a second try:
  - Addition of the `KVM_RUN_X86_GUEST_MODE` flag in `struct kvm_run`
  - Flag indicates the layer of execution
  - Got accepted for Linux 6.11 (just released!)

- Relation between API Changes and root cause:
  - Detect layer of execution
  - Update state according to execution layer



Takeaways: KVM API

- First large KVM project after years of experience with other virtualization APIs
- KVM API is historically grown
  - Need to read KVM/Qemu source code to understand
    - Sprawling interrupt injection API
  - Violates intuitive API assumptions
    - Marking unchanged values as dirty = no-op
    - KVM_GET_LAPIC syncs PIR to IRR non-atomically
- Not clear how to improve – Ideas welcome!
  - Documentation improvements
  - Is there a path to simplifyingg the API?

KVM Backend Status

- It works!
- Compute performance of KVM backend is competitive with Vanilla VBox!
  - Either parity or faster!
  - Legacy IO (SATA) is slower though! (Because it's emulated in vboxdrv.)
- Supports:
  - Tested and stable on AMD and Intel CPUs (with XSAVE support)
  - Nested Virtualization on Intel
- Support VFIO GPU passthrough + virtio-gpu
  - SR-IOV GPU passthrough for Intel Xe GPUs!
- Supports all (almost all) VirtualBox convenience features
  - Guest Integration

VirtualBox KVM Blog Series

We blogged about our VirtualBox KVM adventure.

1. Overview:
   - https://cyberus-technology.de/articles/vbox-kvm-public-release
2. Technical Deep Dive
   - https://cyberus-technology.de/articles/vbox-kvm-deep-dive
3. GPU Virtualization:
   - https://cyberus-technology.de/articles/vbox-kvm-sriov
4. Simple SR-IOV Setup
   - https://cyberus-technology.de/en/articles/simplify-your-sr-iov-setup-a-guide-to-nixos-modules-and-specializations

Summary

- Our patchset allows using VirtualBox without out-of-tree kernel modules!
  - Find it on Github: cyberus-technology/virtualbox-kvm
  - Sponsor us!
- The KVM API is for the brave:
  - Closely tied to Qemu.
  - No way around reading KVM and Qemu source.
  - Happy to talk about improvement paths!
- Packages available!
  - NixOS / Arch / Gentoo
- Commercial Support is available!
  - Virtualization / Nix / NixOS / Testing
  - https://cyberus-technology.de/contact

Company:

- Blog: cyberus-technology.de/articles
- GitHub: github.com/cyberus-technology
- Twitter: @CyberusTech
- Mastodon: https://mstdn.business/@cyberus

Talk to us for virtualization-related consulting!

How to Use It?

We use this daily!

There are packages available:

- ArchLinux AUR: `virtualbox-kvm`
- Gentoo: `app-emulation/virtualbox-kvm`
- NixOS 24.05/unstable: `pkgs.virtualboxKvm`
  - `virtualization.virtualbox.host.enable = true;`
  - `virtualization.virtualbox.host.enableKvm = true;`

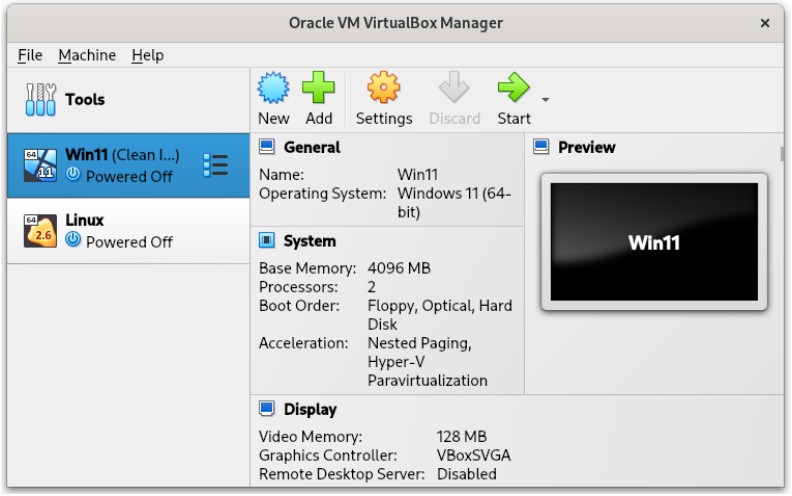
Build instructions in our README on Github:

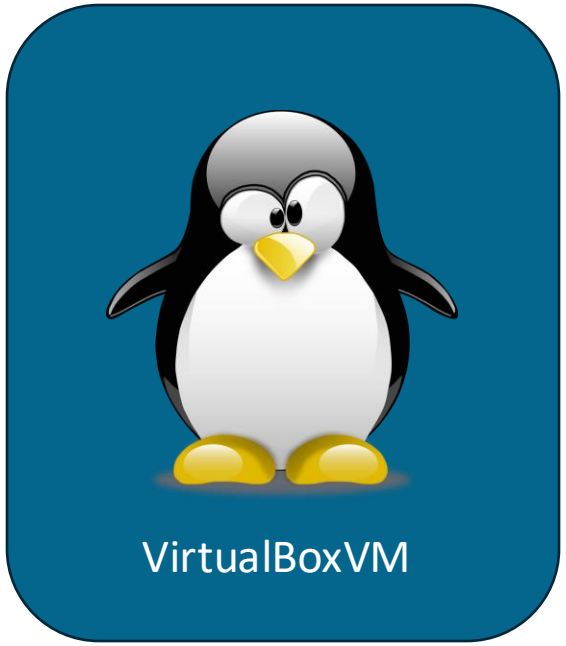
- https://github.com/cyberus-technology/virtualbox-kvm

Some features are incomplete/missing compared to Vanilla Vbox:

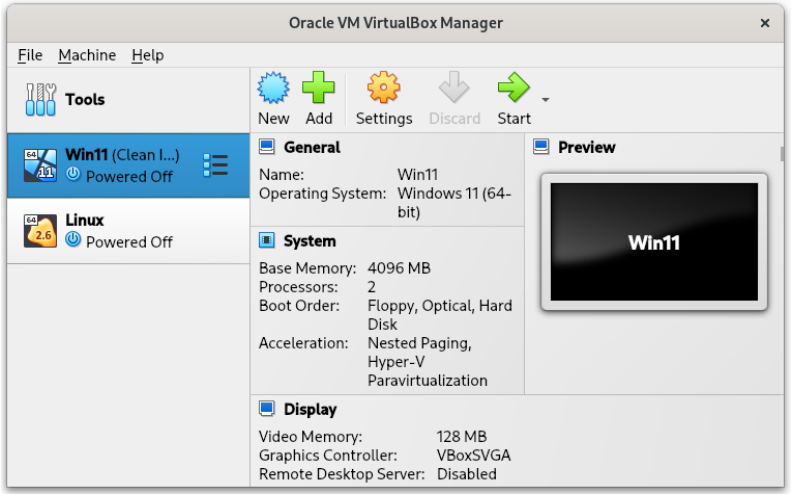
- Advanced Networking

High-Level VirtualBox Architecture





KVM Backend for VirtualBox

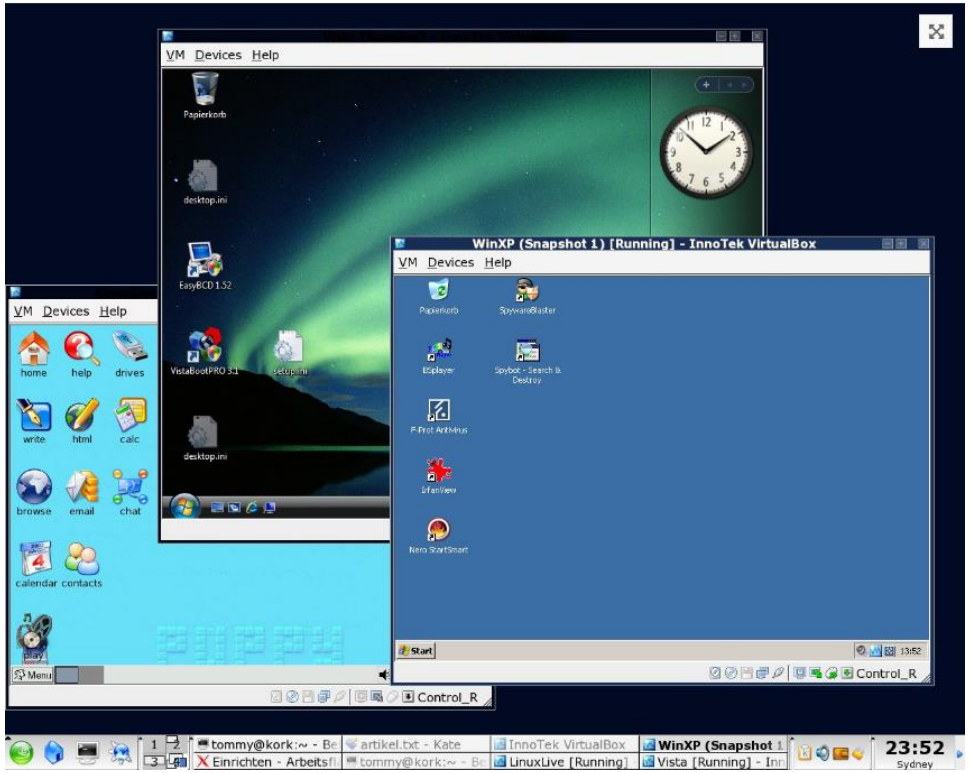


Same user experience!

No 3rd party kmods! 🥳

History of VirtualBox

- Born at InnoTek GmbH in Germany
  - First open source release in 2007
  - First full-featured open source virtualization solution
  - Since then, acquired by Sun and then Oracle
- No mature OS-level Support for Virtualization at the time
  - AMD-V, Intel VT born around the same time!
  - KVM merged into Linux 2.6.20 in 2007
  - MS Hyper-V shipped in 2008
- So VirtualBox just did its own thing.



Source: LinuxUser 04/2007

Open Source Edition vs Extension Pack

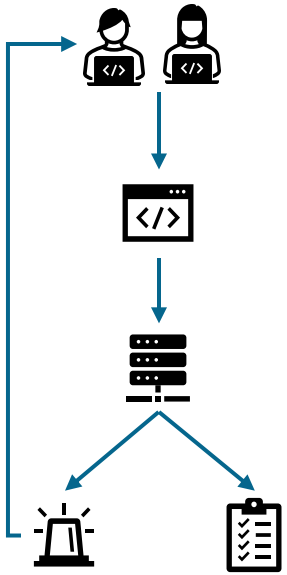
Open Source Edition

- GPLv3 (mostly) with some clarifications
- Kernel modules
  - `vboxdrv` (the actual hypervisor)
  - …
- Qt GUI for configuration and running VMs
- CLI
- Large Feature Set
  - PIIX3/ICH9 Host Bridge
  - Large set of network cards, storage controllers etc
  - BIOS and UEFI boot
  - TPM / Secure Boot
- Hassle-free Windows 11 experience

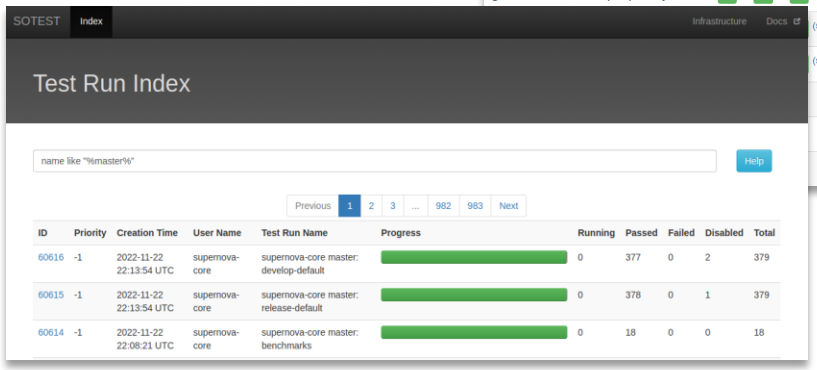
Extension Pack

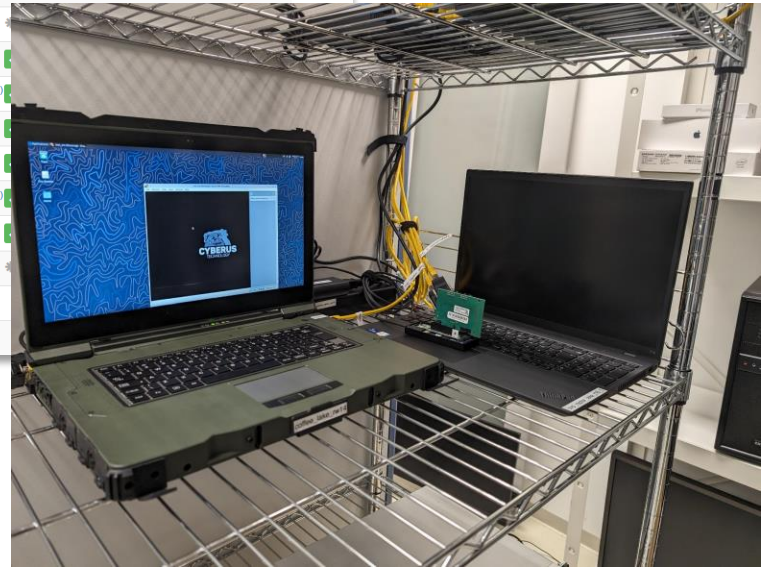
- Plugs into VirtualBox to provide more features
- *Proprietary Software*
  - Free for personal use (check the license!)
- Additional Features
  - Remote Display
  - ~~USB3~~ (moved to OSE!)
  - ~~Webcam Passthrough~~ (moved to OSE!)
  - Guest Network Boot
  - Disk Image Encryption
  - NVMe Emulation ☹
  - …

Avoiding Quality and Performance Regressions



Name	lenovo_sr630	lenovo_sr650v2	lewisburg	sina_h_client_3a	sunrise point	svp_alder_lake_p360	svp_alder_lake_t16	svp_comet_lake_x13	svp_kaby_lake_r_t480s	svp_kaby_lake_x270	svp_sina_1g2	svp_tuxedo_x14_gen12	svp_whiskey_lake_t490	svp_ws_h_dell_wyse_5070	any
app-boot_state	*	*	*	*	*	*	*				*	*		*	✓
app-bss	*	*	*	*	*	*	*				*	*		*	✓
app-delegate	*	*	*	*	*	*	*				*	*		*	✓
app-map_io	*	*	*	*	*	*	*				*	*		*	✓
app-map_memory	*	*	*	*	*	*	*				*	*		*	✓
app-service_call	*	*	*	*	*	*	*				*	*		*	✓
app-simple	*	*	*	*	*	*	*				*	*		*	✓
guest-mini-native-cpuid	✓	(s) ✓	(s) ✓	(s) ✓	(s) ✓	(s) ✓	(s) ✓	(s) ✓	(s) ✓	(s) ✓	(s) ✓	(s) ✓	(s) ✓	(s)	
guest-mini-native-cxx	*	*	*	*	*	*	*	*	*	*	*	*	*	*	✓
guest-mini-native-emulator	*	*	*	*	*	*	*	*	*	*	*	*	*	*	✓
guest-mini-native-emulator-syscall	*	*	*	*	*	*	*								
guest-mini-native-exceptions	✓	✓	✓	✓	✓	✓	✓								
guest-mini-native-fpu	✓	(s) ✓	(s) ✓	(s) ✓	(s) ✓	(s) ✓	(s) ✓	(s) ✓							
guest-mini-native-lapic-modes	✓	✓	✓	✓	✓	✓	✓								
guest-mini-native-lapic-priority	✓	✓	✓	✓	✓	✓	✓								

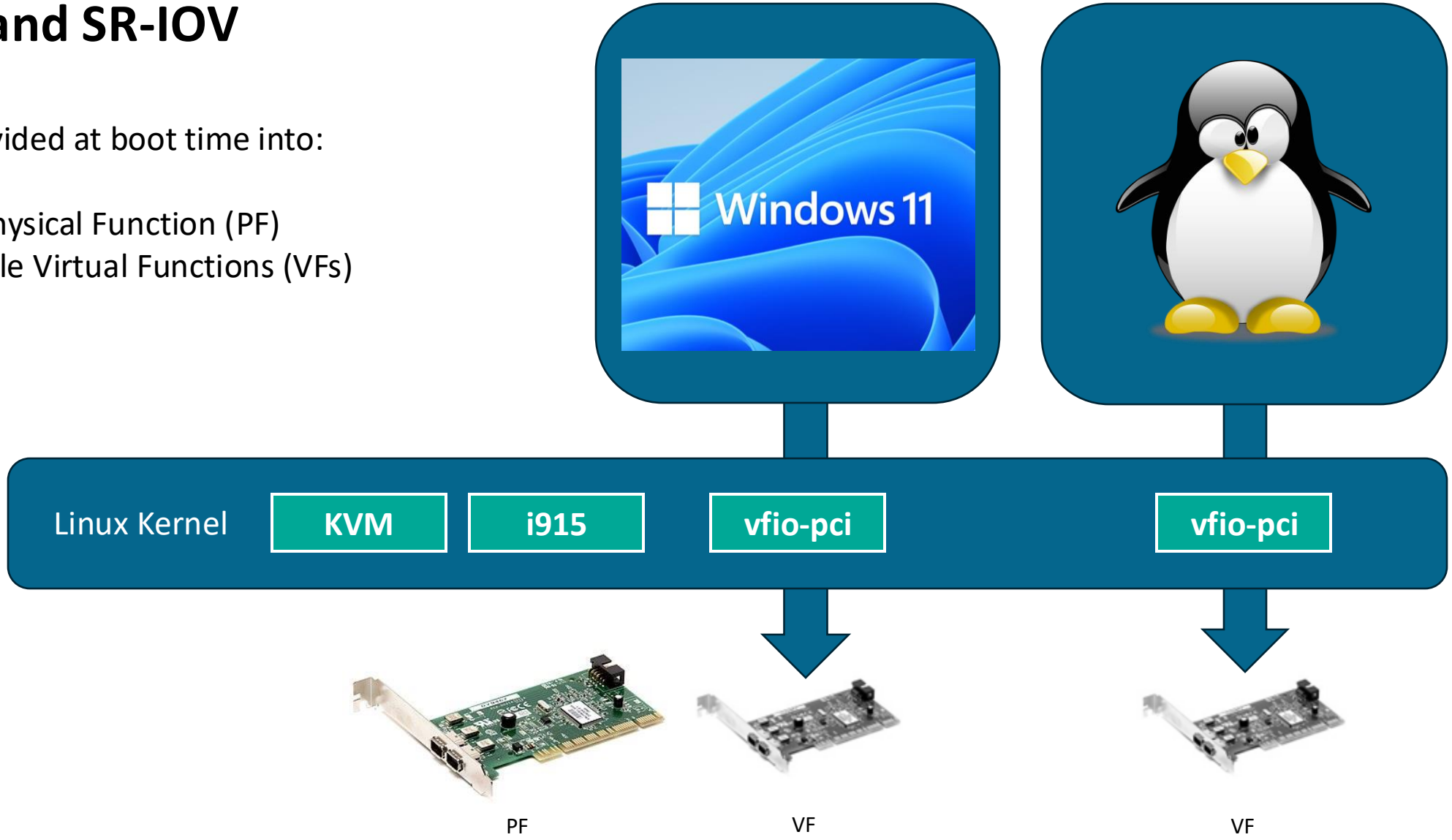




GPUs and SR-IOV

GPU is divided at boot time into:

- One Physical Function (PF)
- Multiple Virtual Functions (VFs)

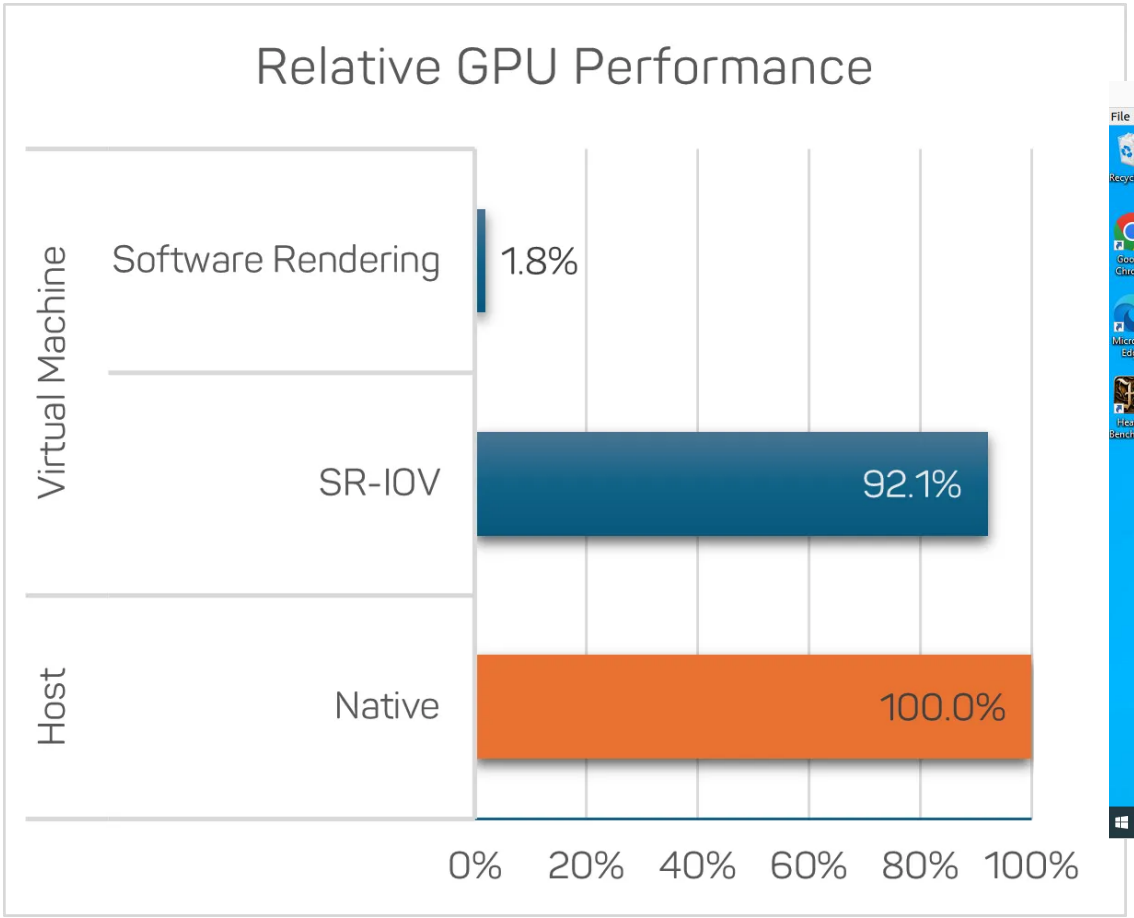


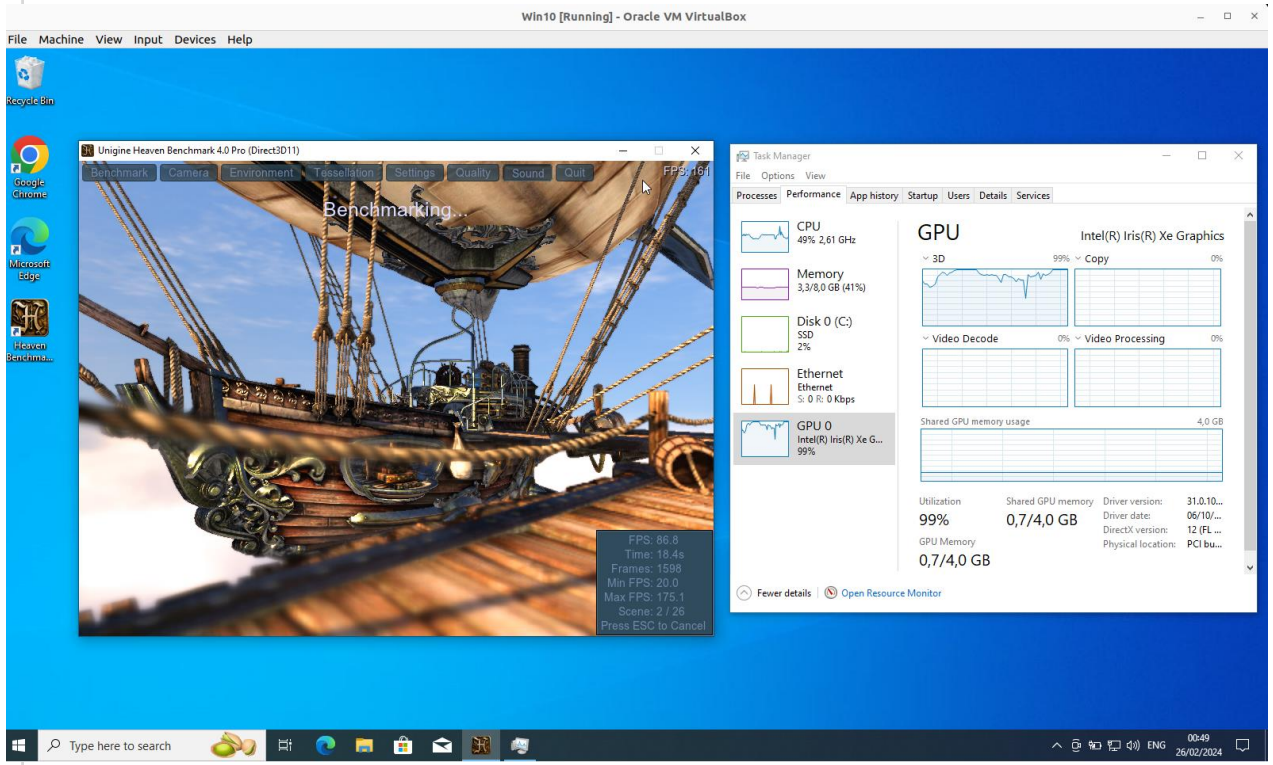
GPU Virtualization: Availability

Available from Intel 12th Core processors onwards.

```
❯ sudo lspci -v -s 00:02.0
00:02.0 VGA compatible controller: Intel Corporation AlderLake-UP3 GT2 [Iris Xe Graphics]
…
  Capabilities: [320] Single Root I/O Virtualization (SR-IOV)
  Kernel driver in use: i915
  Kernel modules: i915
```

GPU Virtualization





Details: https://cyberus-technology.de/articles/vbox-kvm-sriov

Upstreaming Status: Intel Xe Driver

- Required features are not in Intel i915 driver
  - Need to use Intel Linux 6.6 LTS branch
- Intel works on a new driver for new hardware
  - Designed for Tiger Lake and up
  - Might be enabled by default for Intel Lunar Lake?
  - Will carry SR-IOV code
- Experimental version ships in Linux 6.8
  - Not enabled by default

Performance

Geekbench Multi Core

Machine Name	Windows 11 Overhead	Windows 10 Overhead
svp_whiskey_lake_t490	1.04	1.03
svp_raptor_lake_t16	1.02	1.01
svp_kaby_lake_r_t480s	0.799	0.796
svp_comet_lake_x13	1.03	1.03
svp_alder_lake_t16	0.994	0.996
svp_alder_lake_p360	0.999	1.00
coffee_lake_rw14	0.796	0.804

Disk Speed SEQ128K Q32T1 RD

Machine Name	Windows 11 SATA	Windows 10 SATA	Windows 11 NVMe	Windows 10 NVMe
svp_whiskey_lake_t490	0.990	0.994	0.997	0.998
svp_raptor_lake_t16	1.16	1.09	0.974	0.962
svp_kaby_lake_r_t480s	0.865	0.875	0.991	0.986
svp_comet_lake_x13	0.997	0.995	1.00	1.00
svp_alder_lake_t16	1.11	1.10	0.990	0.957
svp_alder_lake_p360	0.998	0.999	0.999	0.999
coffee_lake_rw14	1.01	1.02	0.992	0.996

Disk Speed RND4K Q1T1 RD

Machine Name	Windows 11 SATA	Windows 10 SATA	Windows 11 NVMe	Windows 10 NVMe
svp_whiskey_lake_t490	1.29	1.25	0.939	0.937
svp_raptor_lake_t16	1.06	1.07	0.750	0.748
svp_kaby_lake_r_t480s	1.22	1.20	0.929	0.921
svp_comet_lake_x13	1.25	1.24	0.956	0.957
svp_alder_lake_t16	1.11	1.09	0.728	0.715
svp_alder_lake_p360	1.13	1.12	0.686	0.691
coffee_lake_rw14	1.37	1.33	0.992	0.967

Performance