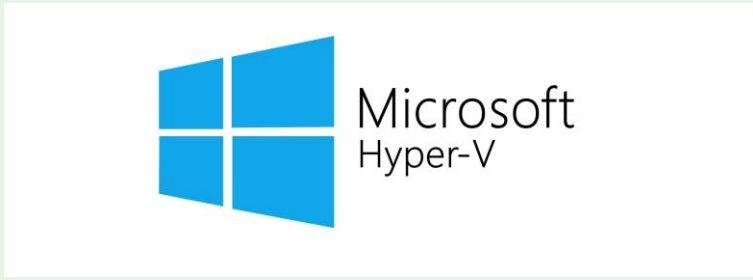
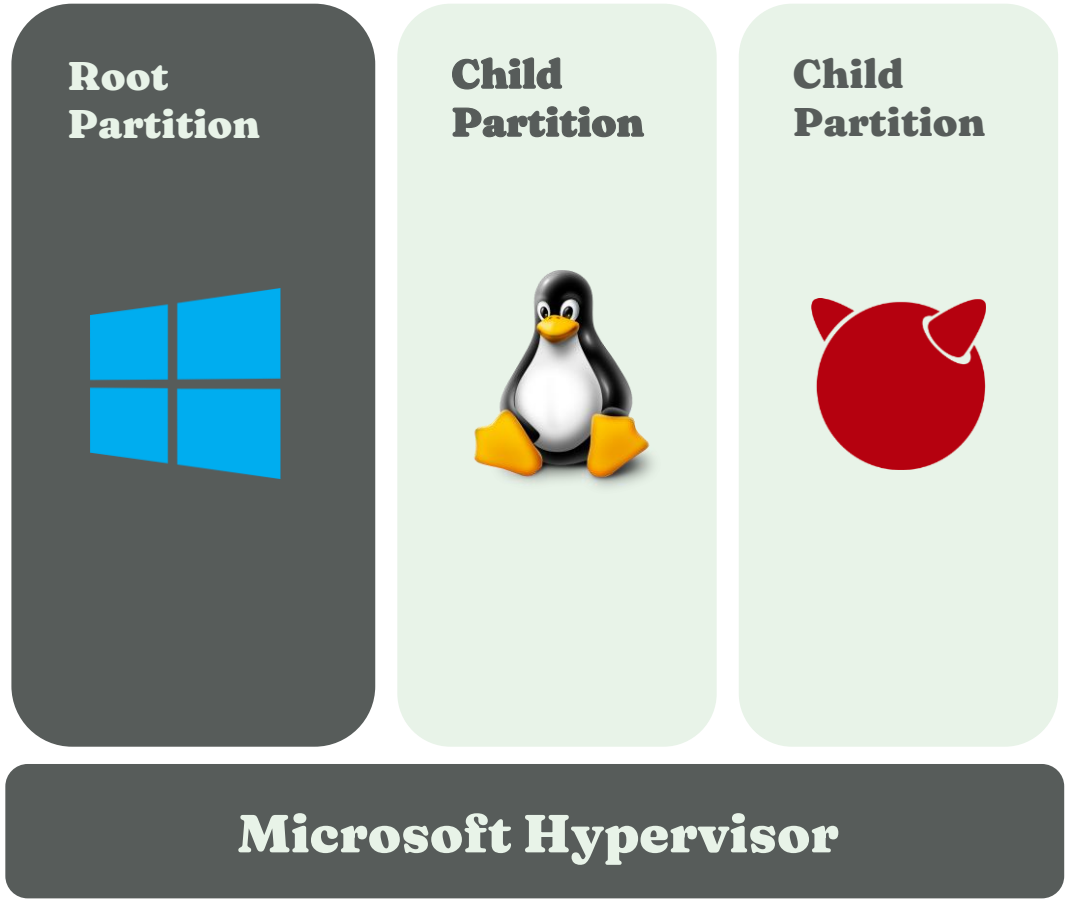
Beneath the Surface: Analyzing Nested CVM Performance on KVM/QEMU and Linux Root Partition for Microsoft Hyper-V/Cloud-Hypervisor

Muminul Islam
Jinank Jain
Linux Systems Group - Microsoft

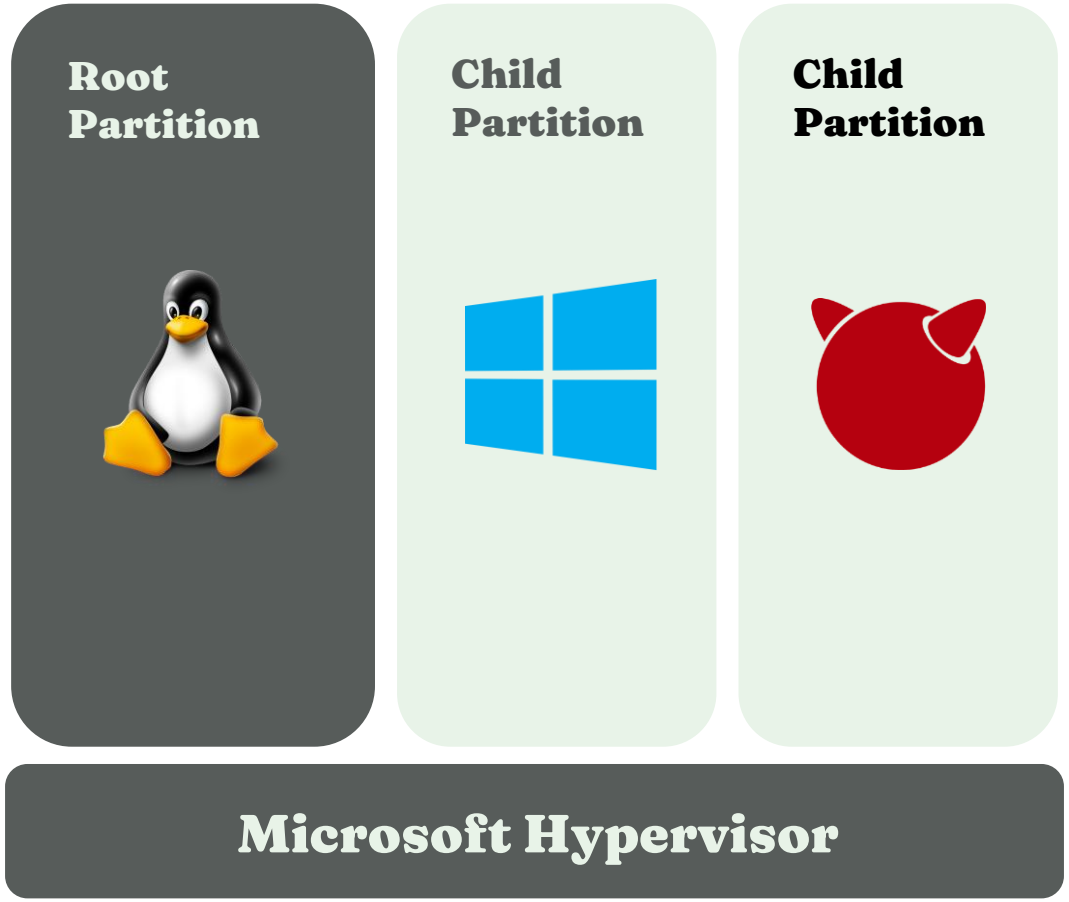


Microsoft Hypervisor Virtualization Stack

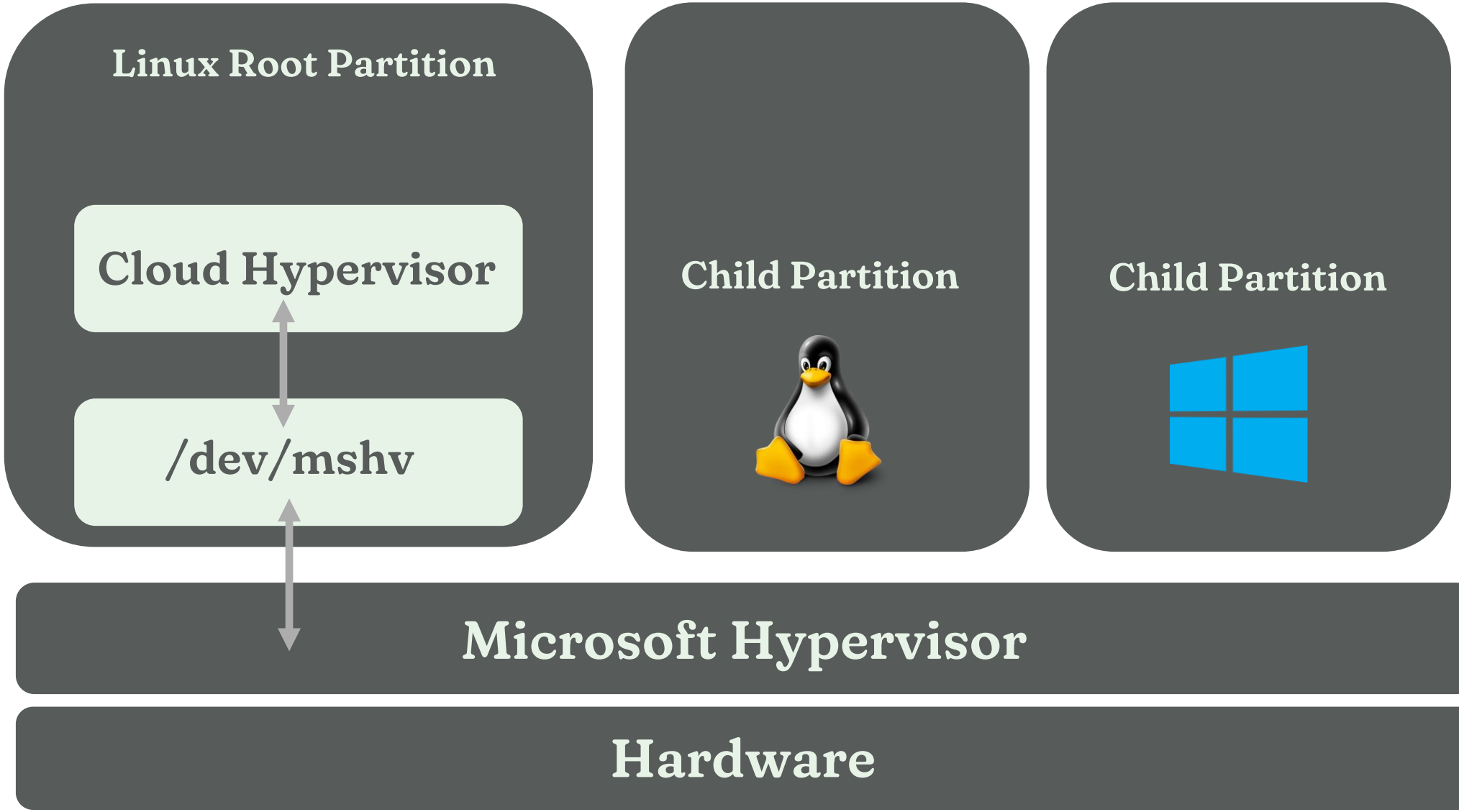
Windows Root Partition



Linux Root Partition

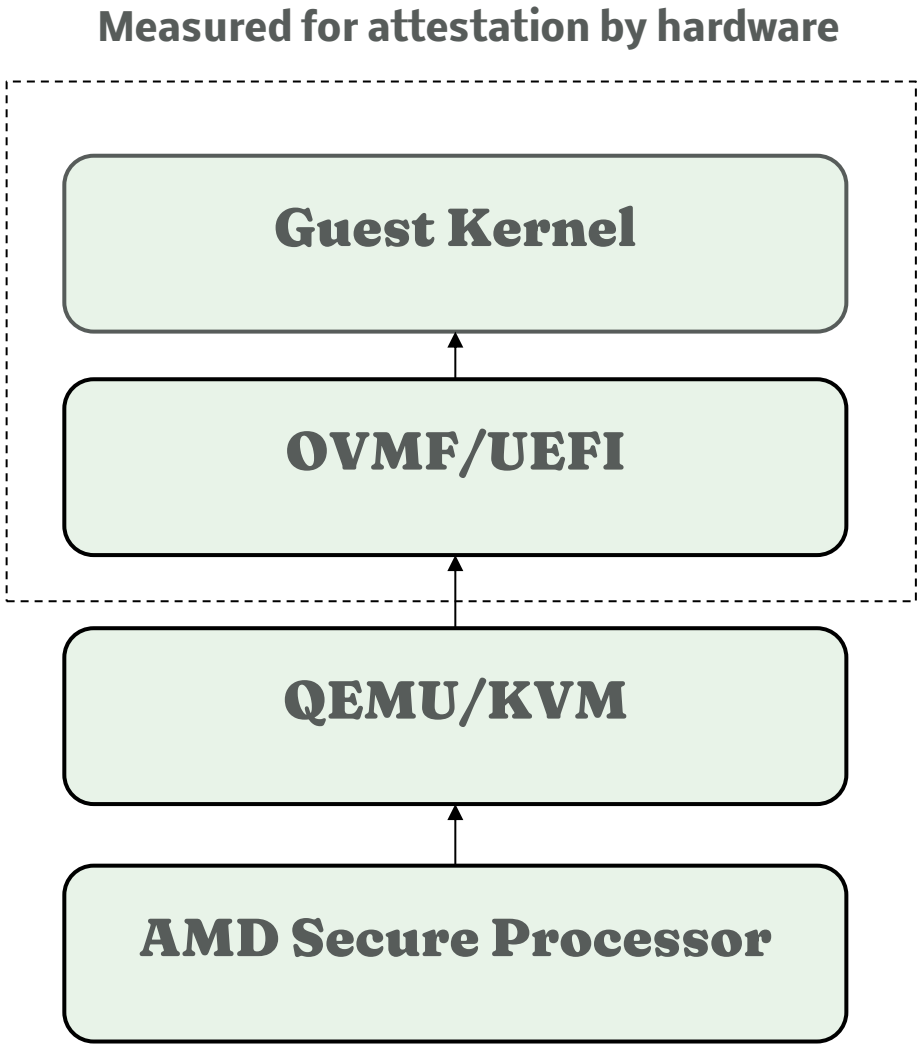


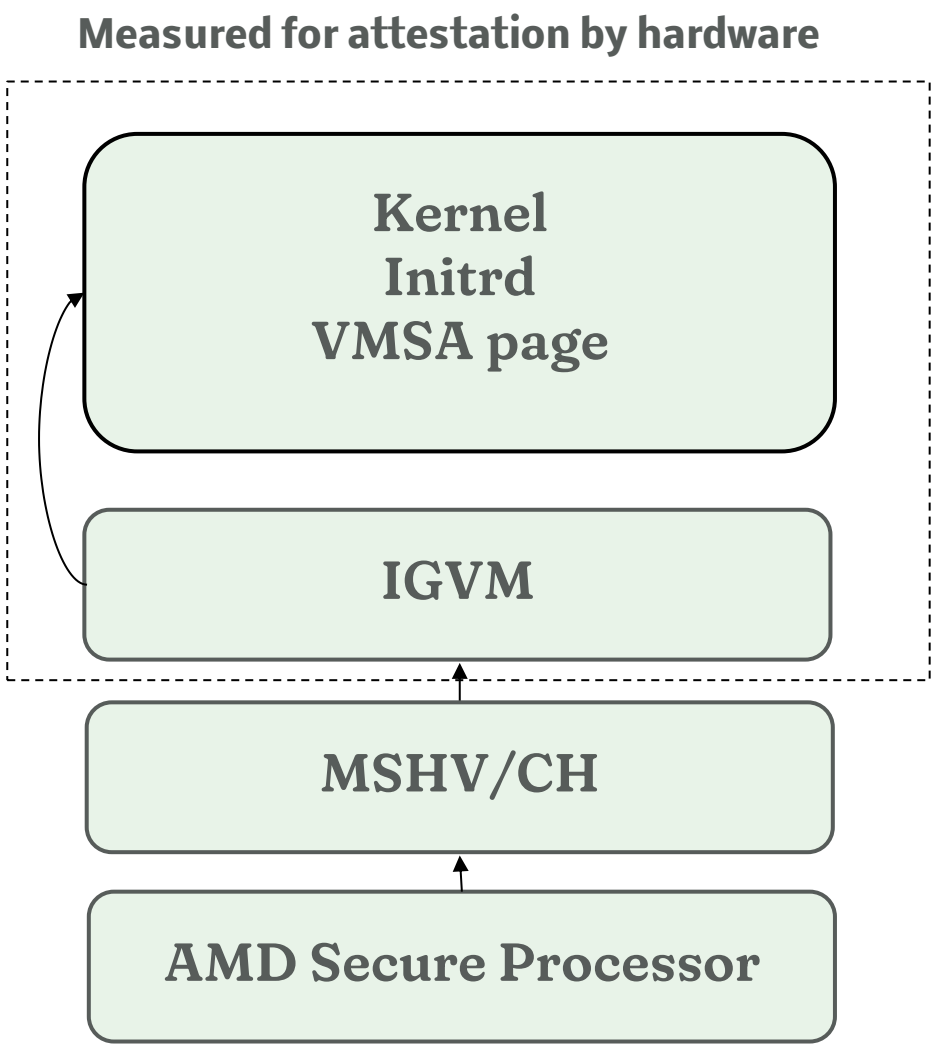
Linux MSHV Virtualization Stack



LKML: https://lore.kernel.org/lkml/1692309711-5573-1-git-send-email-nunodasneves@linux.microsoft.com/

Confidential VMs on KVM/QEMU vs MSHV/CH





Independent Guest Virtual Machine (IGVM)

- A virt stack agnostic way to package the launch information for a guest VM into a single file
- Consists of an ordered list of directives with accompanying data specifying how the virt stack should load the guest
  - Note the ordering allows us to create stable, signed launch measurements for SNP, etc.
- Think of it like ELF/PE for VM launch
- Open source format published on Github and crates.io:
  - igvm_defs - crates.io: Rust Package Registry
  - igvm - crates.io: Rust Package Registry

CloudHypervisor changes to support CVM

Cloud-Hypervisor changes

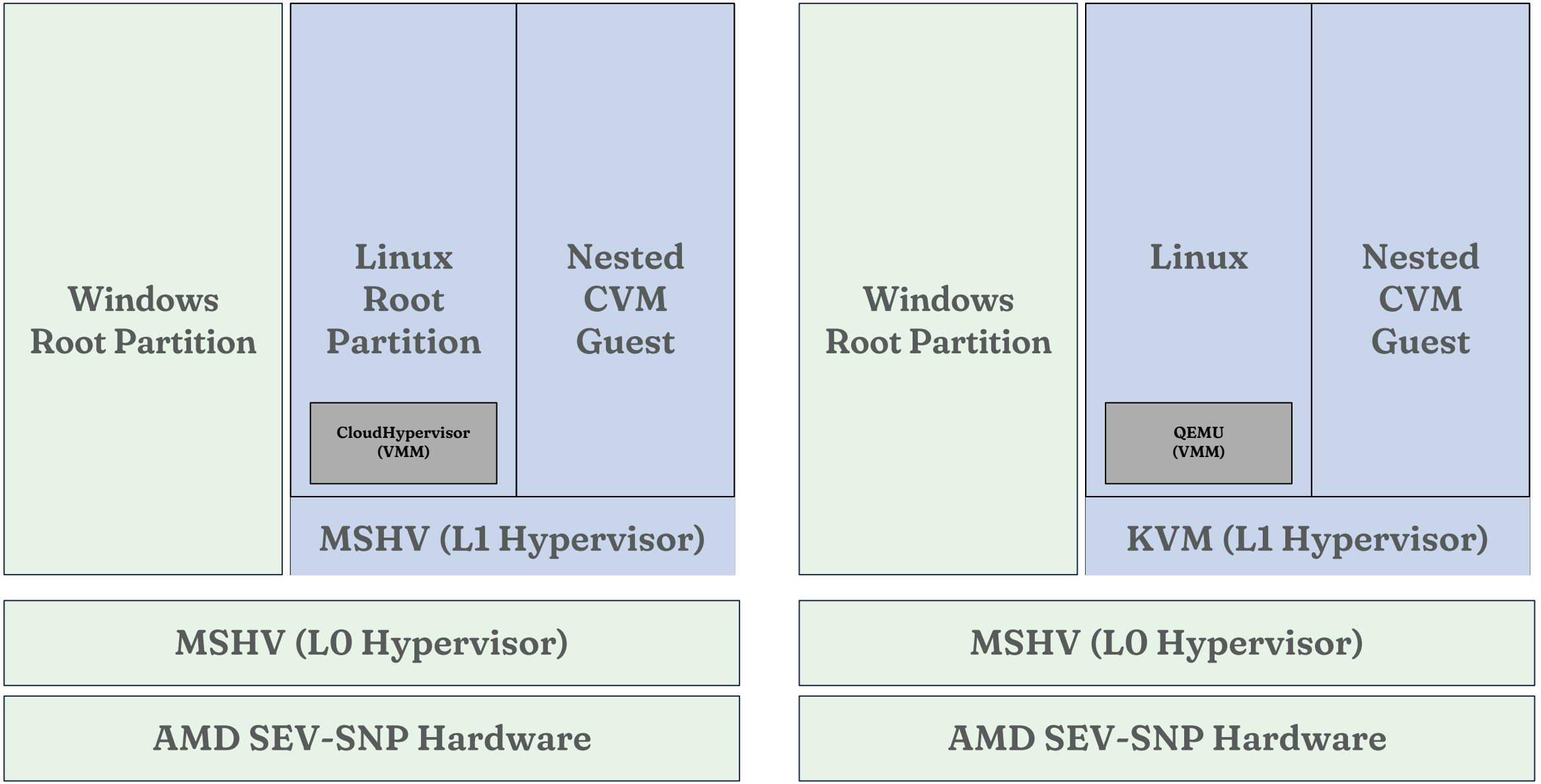
- **IGVM parser**
  - Build on top of open-source crates like IGVM
  - Parse the IGVM file and load into guest memory
  - Performs SEV-SNP specific operations like handling CPUID, secrets pages etc.
- **GHCB exit handling**
  - MSHV specific VCPU run adaptation
  - GHCB/DoorBell page registration
  - MMIO, IoPort exits handling

Cloud-Hypervisor changes, Cont..

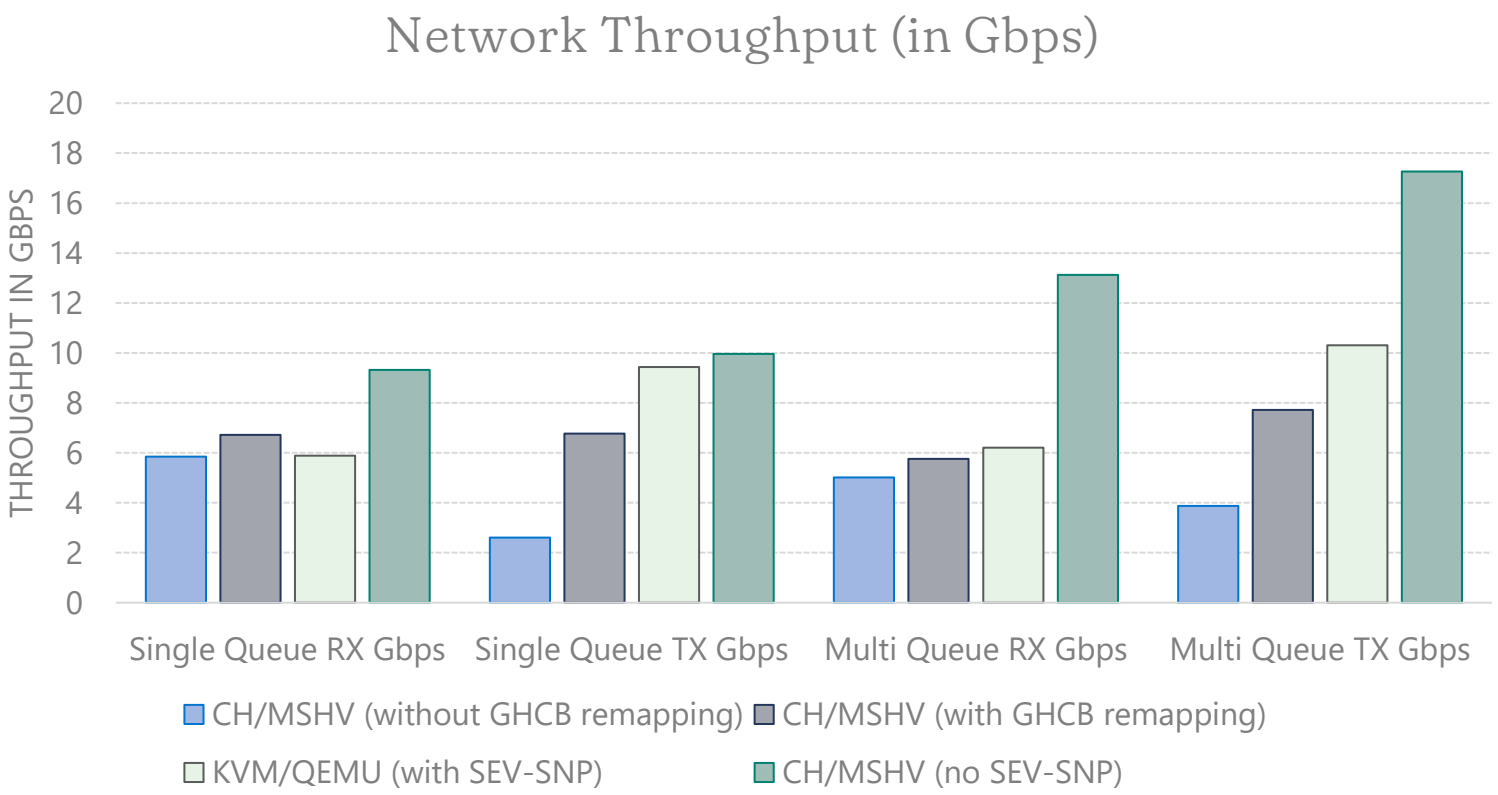
- **Virtio-thread adaptations**
  - Pages are not accessible for VMM to process IO
  - VMM requests hypervisor to access the pages
  - Caching of accessed pages
  - Revocation of the pages from the cache

Performance comparison between Linux/KVM/QEMU vs Linux/MSHV/CloudHypervisor

Performance test setup



Network Throughput

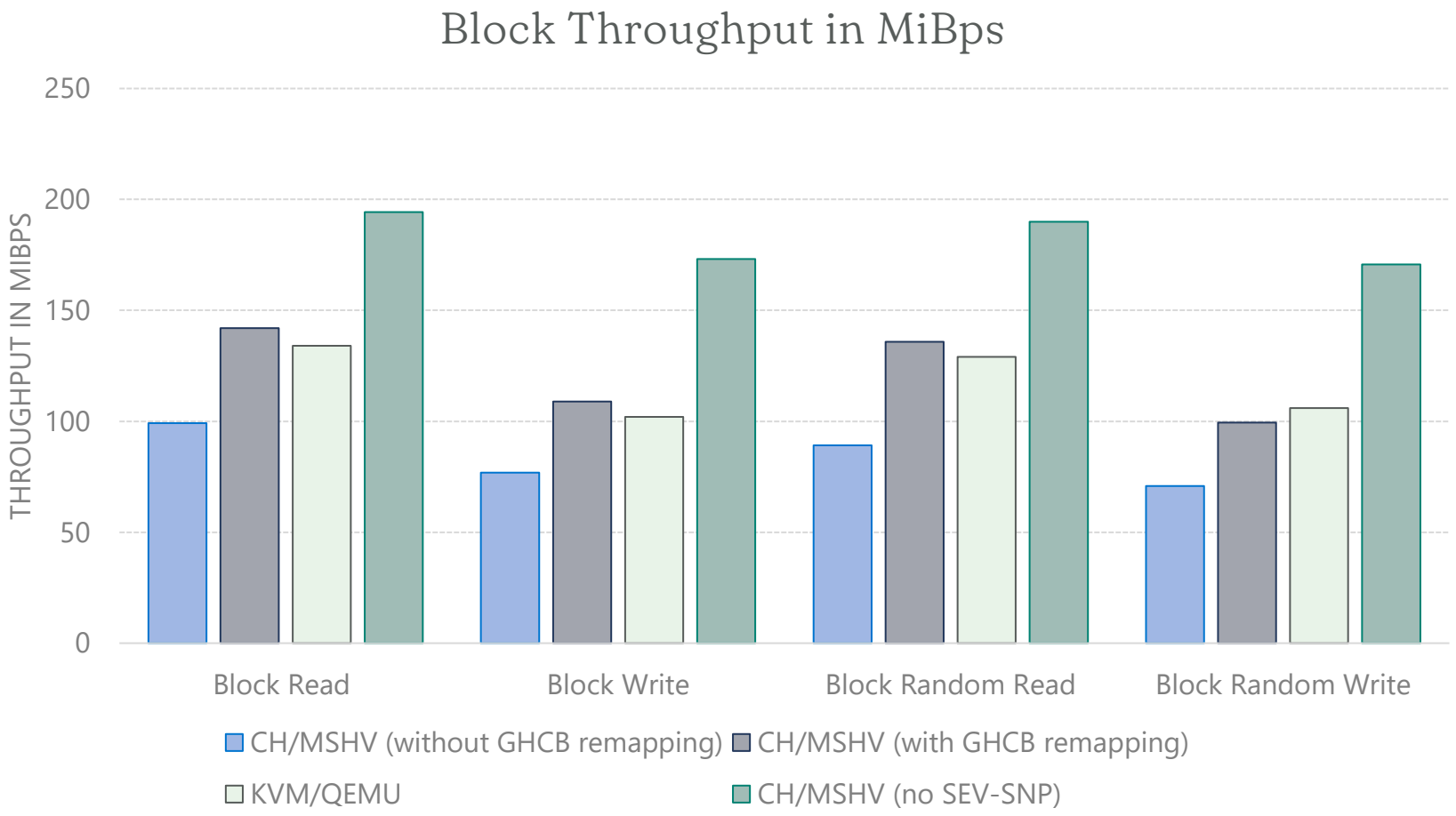


- ## Hardware Setup
  - Processor: AMD EPYC 7763 64-Core Processor 2.45 GHz
  - Installed RAM: 1.00 TB
  - L1 VM: 16 CPU and 32 GiB RAM
- ## Single Queue Setup
  - L2 VM: 1 CPU and 4 GiB
- ## Multi Queue Setup
  - L2 VM: 8 CPU and 8 GiB

Re-mapping GHCB in VMM

- **Problem without re-mapping**
  - Addition IOCTL and HyperCalls for each GHCB page access
  - Performance degradation
- **GHCB remapping**
  - Map GHCB Overlay page of the hypervisor into the VMM process
  - Got rid of context switch to update GHCB page via hypercall
  - Direct read/write GHCB page from the VMM
  - Improved performance

Block I/O Throughput



Hardware Setup

Processor: AMD EPYC 7763 64-Core

Processor 2.45 GHz

Installed RAM: 1.00 TB

L1 VM: 16 CPU and 32 GiB RAM

L2 VM: 8 CPU and 8 GiB

Disk Size: 4GB

CPU Stress Test

$ sysbench --test=cpu --cpu-max-prime=20000 run

KVM/QEMU Guest:

Number of threads: 1

Prime numbers limit: 20000

CPU speed:

events per second: 621.41

General statistics:

total time: 10.0021s

total number of events: 6216

Latency (ms):

min: 0.92

avg: 1.61

max: 4.13

95th percentile: 1.70

sum: 9995.15

Threads fairness:

events (avg/stddev): 6216.0000/0.00

execution time (avg/stddev): 9.9951/0.00

MSHV/CH Guest:

Number of threads: 1

Prime numbers limit: 20000

CPU speed:

events per second: 699.22

General statistics:

total time: 10.0013s

total number of events: 6994

Latency (ms):

min: 0.92

avg: 1.43

max: 6.03

95th percentile: 1.64

sum: 9994.40

Threads fairness:

events (avg/stddev): 6994.0000/0.00

execution time (avg/stddev): 9.9944/0.00

Discussion Points/Future Work

- Measurement of ACPI tables in guest attestation report
- Alternative CVM-native firmware like td-shim
- Boot time optimizations - hashing rate is 2ms per page, large guest take a lot of time to boot
- Live migration support for AMD SEV-SNP CVM on CloudHypervisor

Q & A?