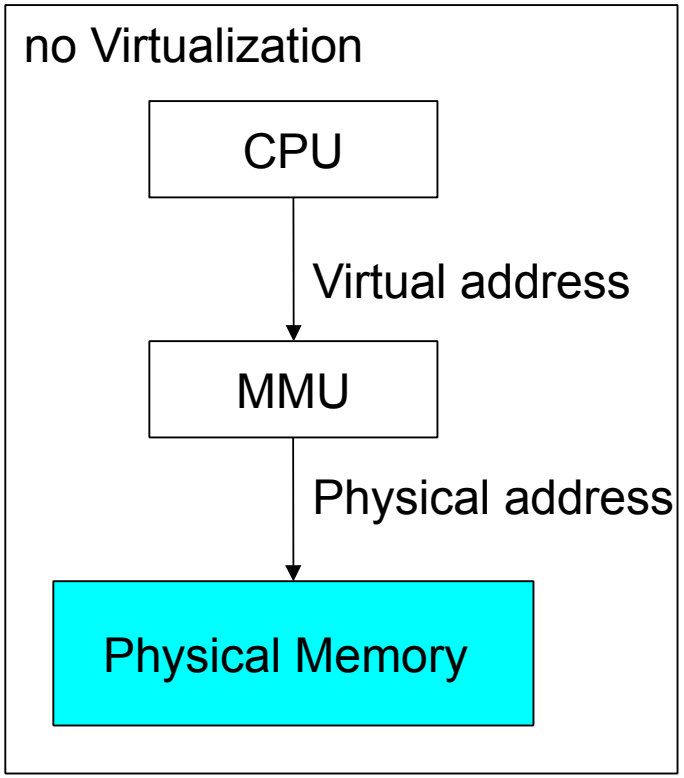
# KVM MMU Virtualization

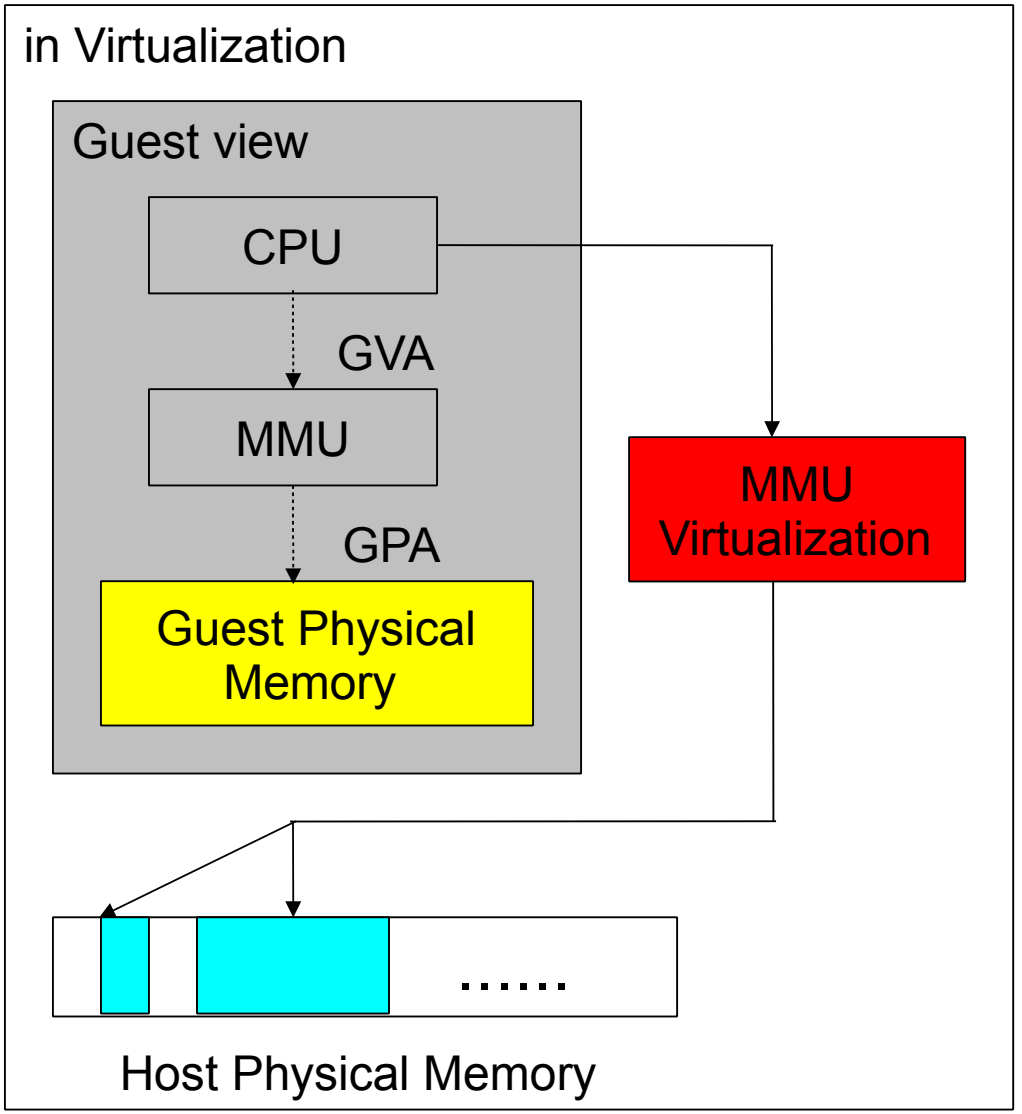
Xiao Guangrong
<xiaoguangrong@linux.vnet.ibm.com>

# Index

- Overview
- Soft MMU
- Hard MMU
- Nested MMU

# Overview



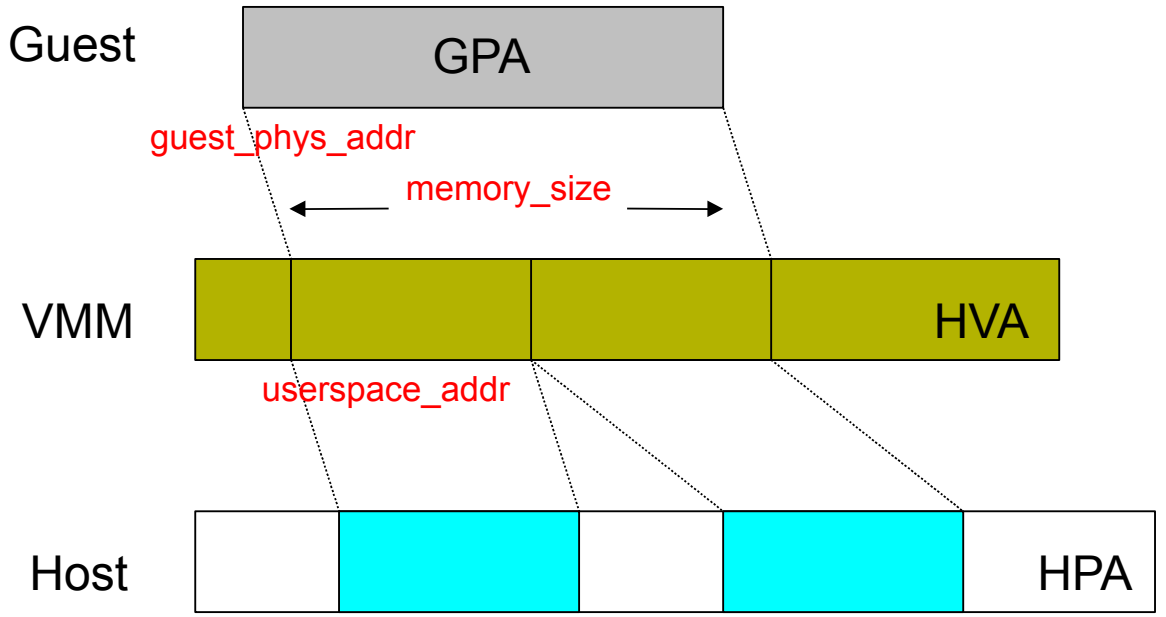

GVA: guest virtual address
GPA: guest physical address

# The functions of MMU Virtualization

- Translate guest physical memory to the specified host physical memory
- Control the memory access permission
  - R/W, NX, U/S
- Track Accessed/Dirty bits of guest page table

# GFN to PFN in KVM

- Use ioctl(fd, KVM_SET_USER_MEMORY_REGION, kvm_userspace_memory_region) to register guest physical memory
  - guest_phys_addr, memory_size, userspace_addr

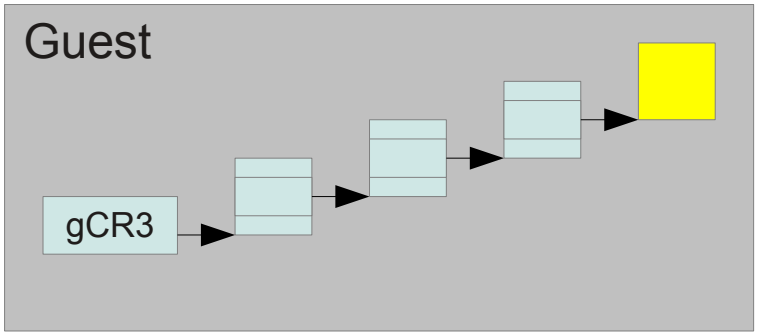


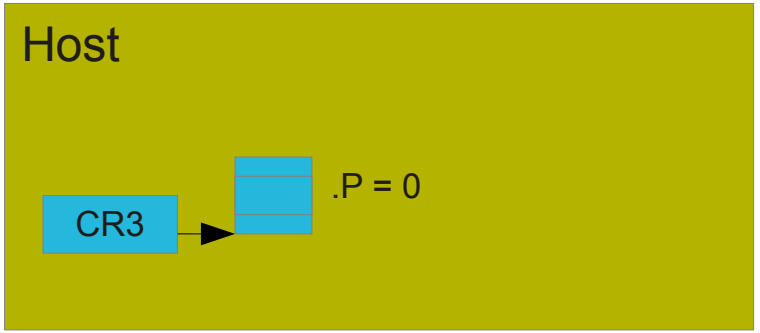
# Soft MMU

- Implemented by software, also known as shadow page table
- Host offers shadow page tables to translate GVA to HPA
- Implementation
  - Initialization
  - Establishment
  - Synchronization

# Soft MMU: Initialization

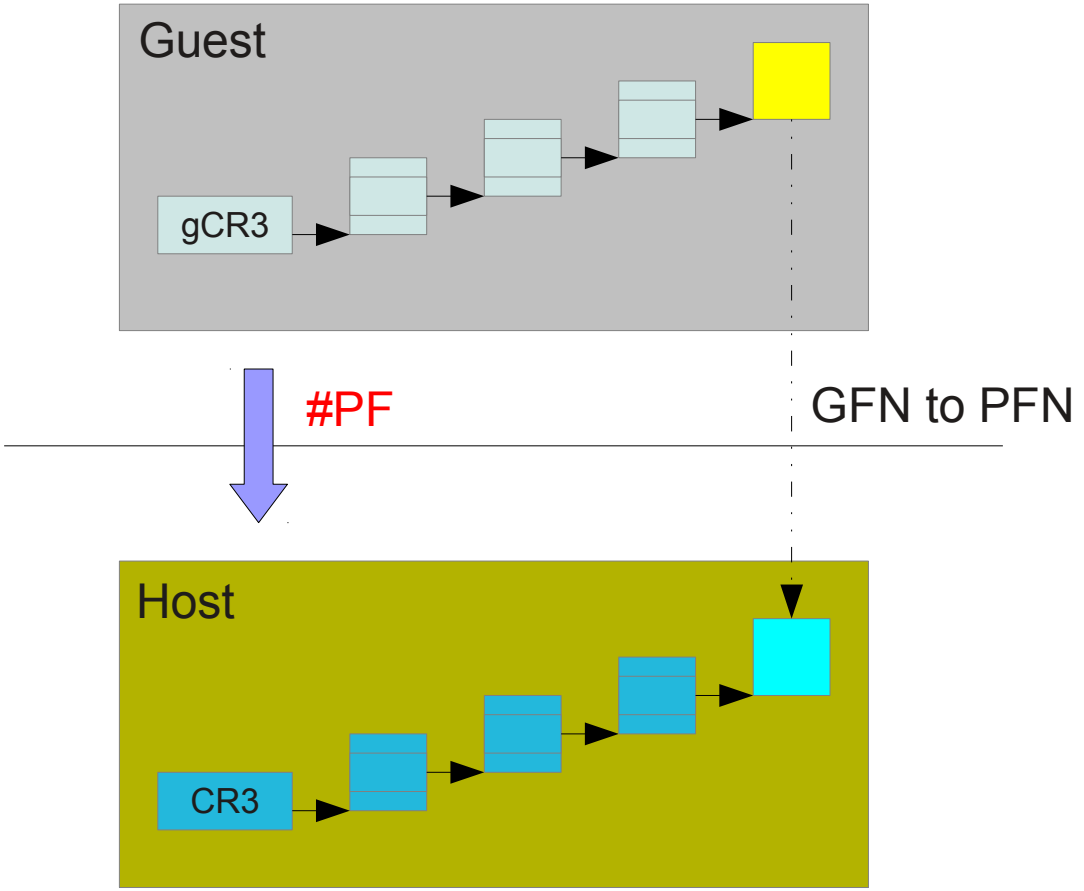
- The shadow page table is empty



# Soft MMU: Establishment

- Intercepting #PF when guest accesses memory, host establishes shadow page table



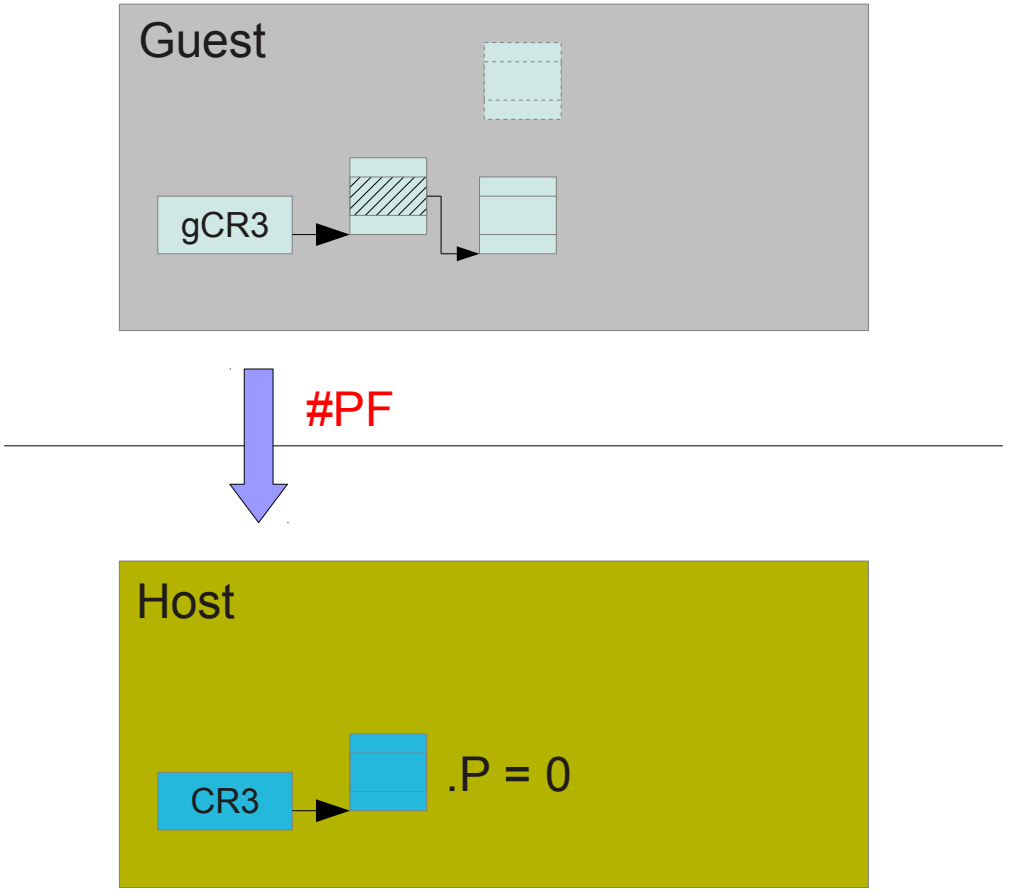

#PF: page fault

# Soft MMU: synchronization (1)

- Track Guest MMU events
  - Paging mode (CR0, CR4 ... ), TLB Flush, #PF, ...
- Track changes to page table in the guest
  - Write-protect the pages containing PML4/PDPT/PDT/PT of the guest
- Track guest dirty pages
  - Write-protect guest pages that are marked clean in the guest
- Explicit sync
  - Special case: sometimes its okay to have the lowest level page mapping to be out of sync. Flush TLB will notify the host which can then update its shadow page table with the latest contents of the guest page table
- Track guest page table Read/Switch
  - Intercept load/store CR3

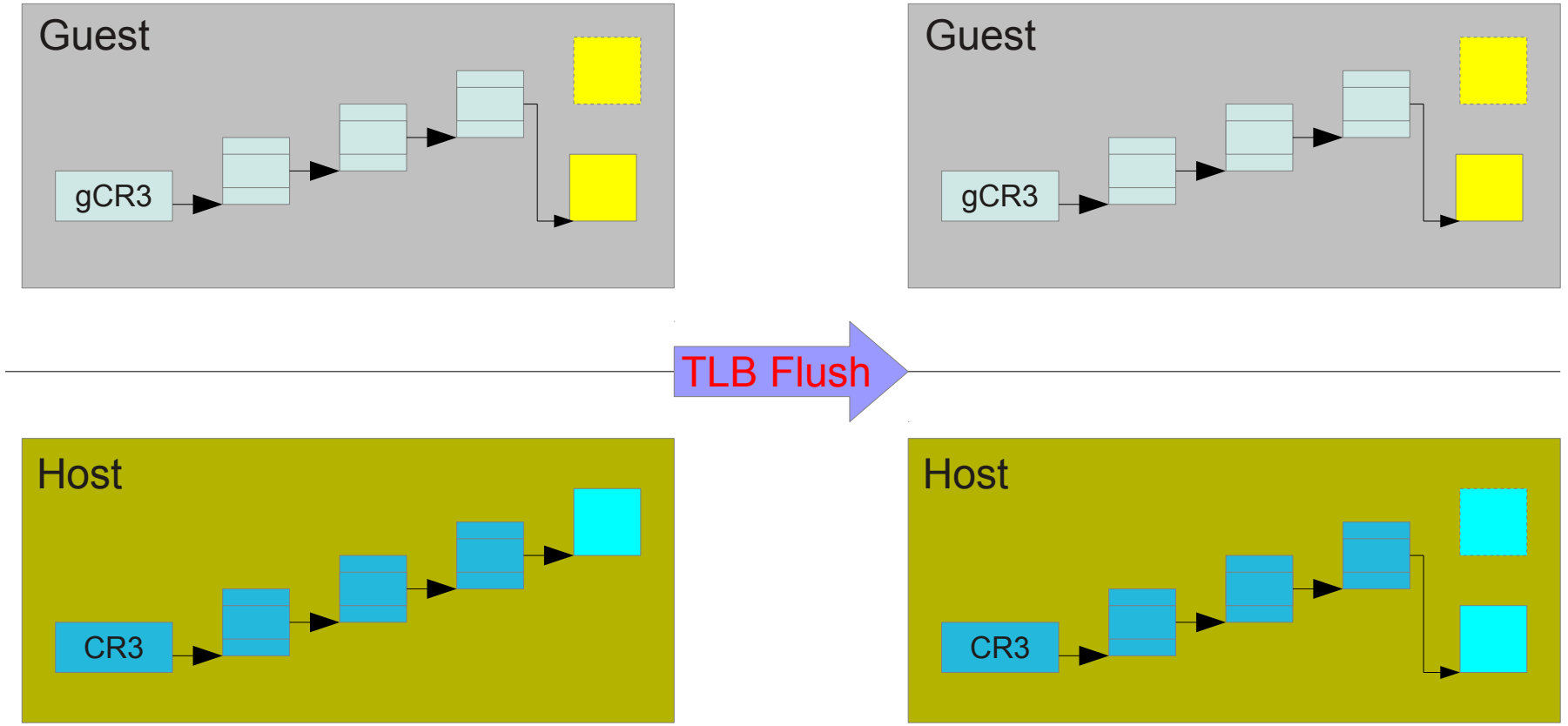
# Soft MMU: Synchronization (2)

- The paging-structure is write-protected, so host can intercept the change of address mapping



# Soft MMU: Establishment (3)

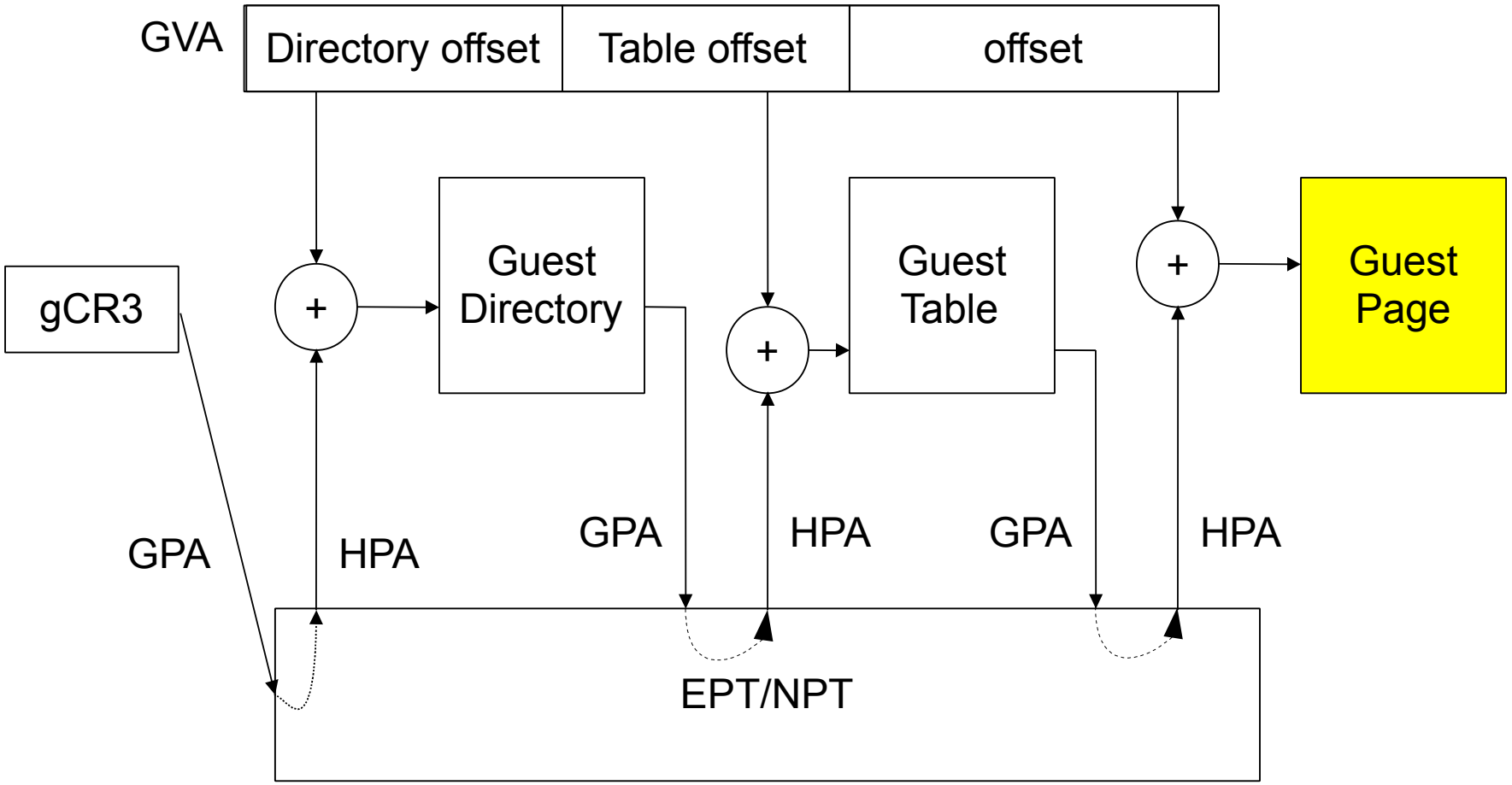
- Special case – unsync shadow page, only allowed on L1 paging-structures,  be sync-ed when TLB is being flushed



# Hard MMU

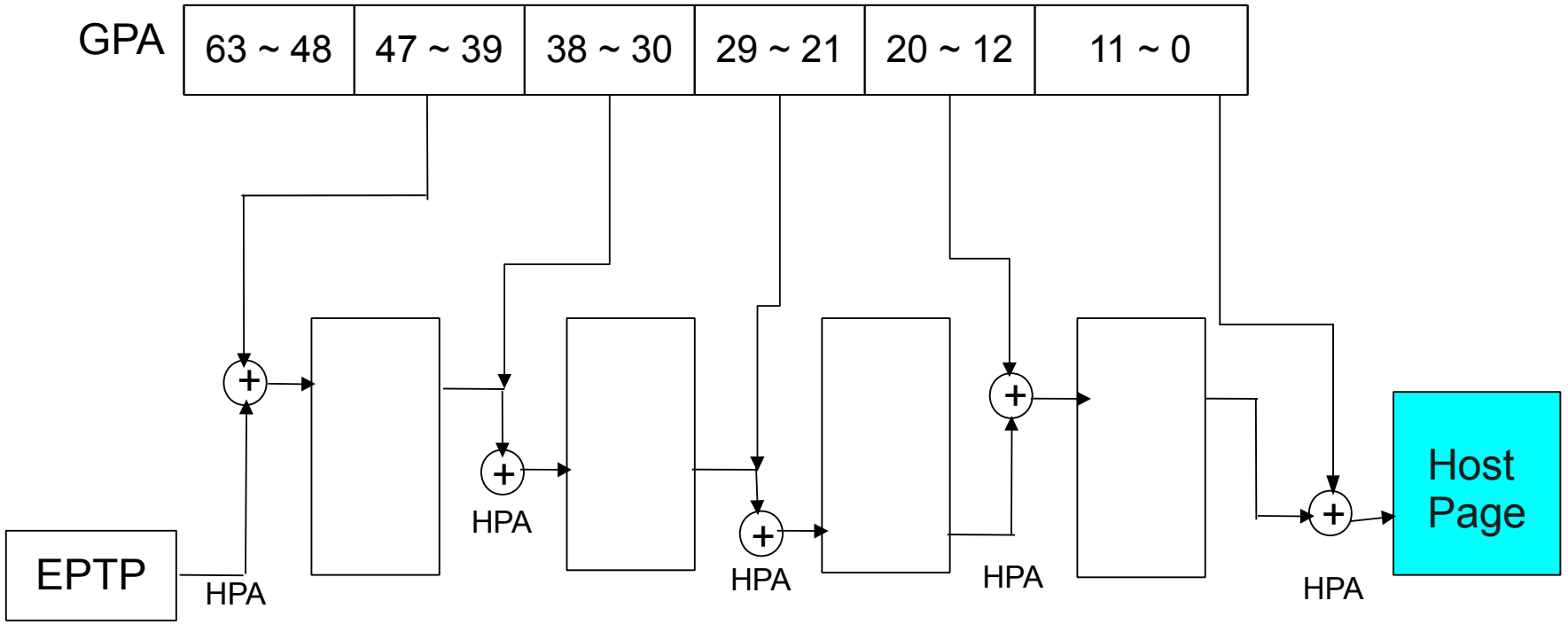
- The address translation is supported by hardware
  - EPT on Intel CPU / NPT on AMD CPU
- Functions
  - The new layer to translate guest physical address to host physical address
  - Use EPT/NPT for all guest physical address access, including MMIO and guest page table walking
  - EPT Misconfig or Violation / #NPF is generated if EPT/NPT page table is invalid
- Comparing to Soft MMU
  - It is simple
  - Need not care the events of guest MMU...

# Hard MMU: overview



# Hard MMU: translate GPA to HPA

EPT/NPT

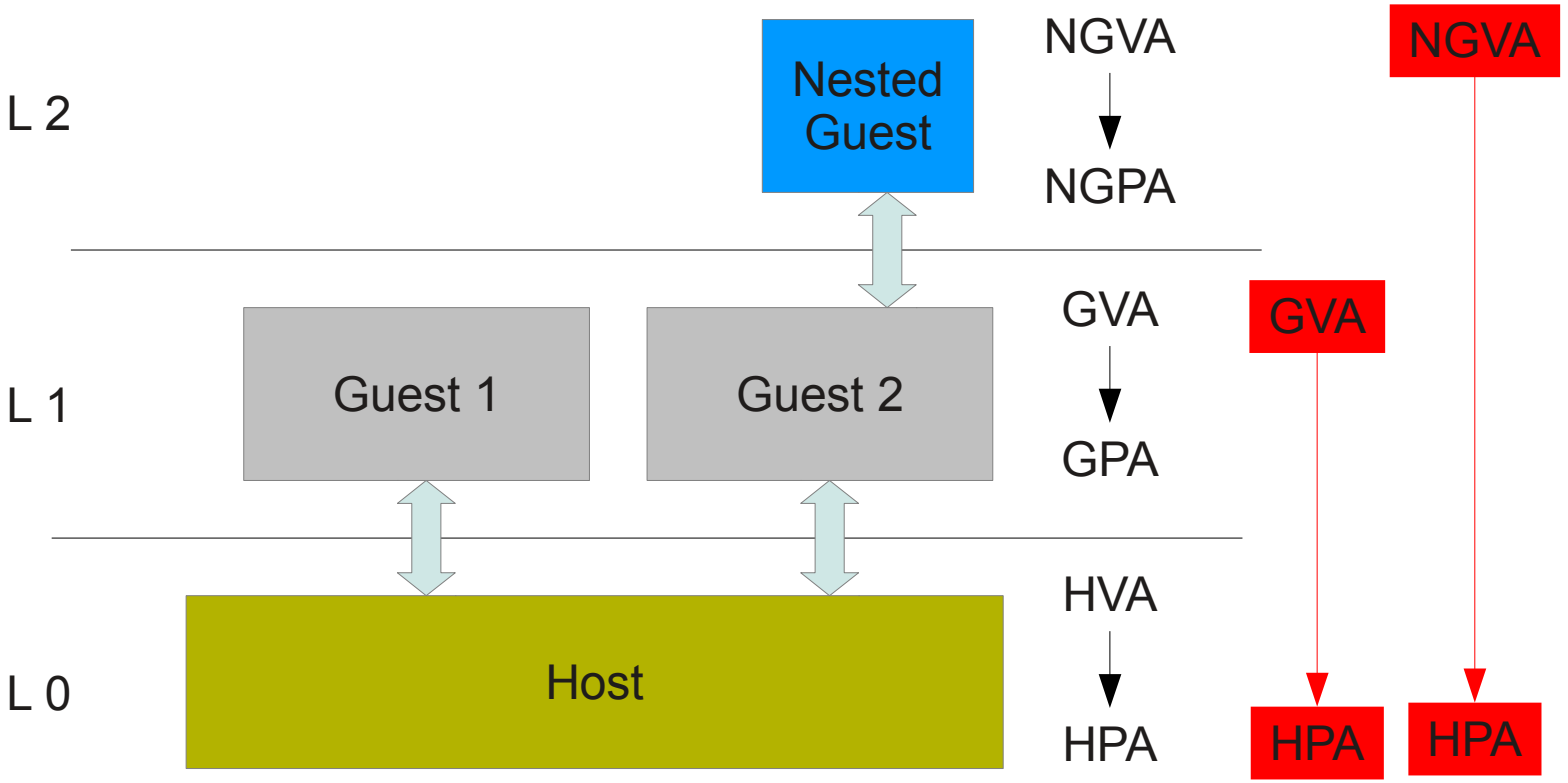


# Nested MMU

- Nested guest
  - Run KVM guests on a KVM guest
- Nested MMU
  - MMU Virtualization on Nested guest
- Implementation
  - Soft MMU on Soft MMU
  - Soft MMU on Hard MMU
  - Hard MMU on Hard MMU
- Take EPT as a example in the follow descriptions

# Nested MMU: overview

- Overview



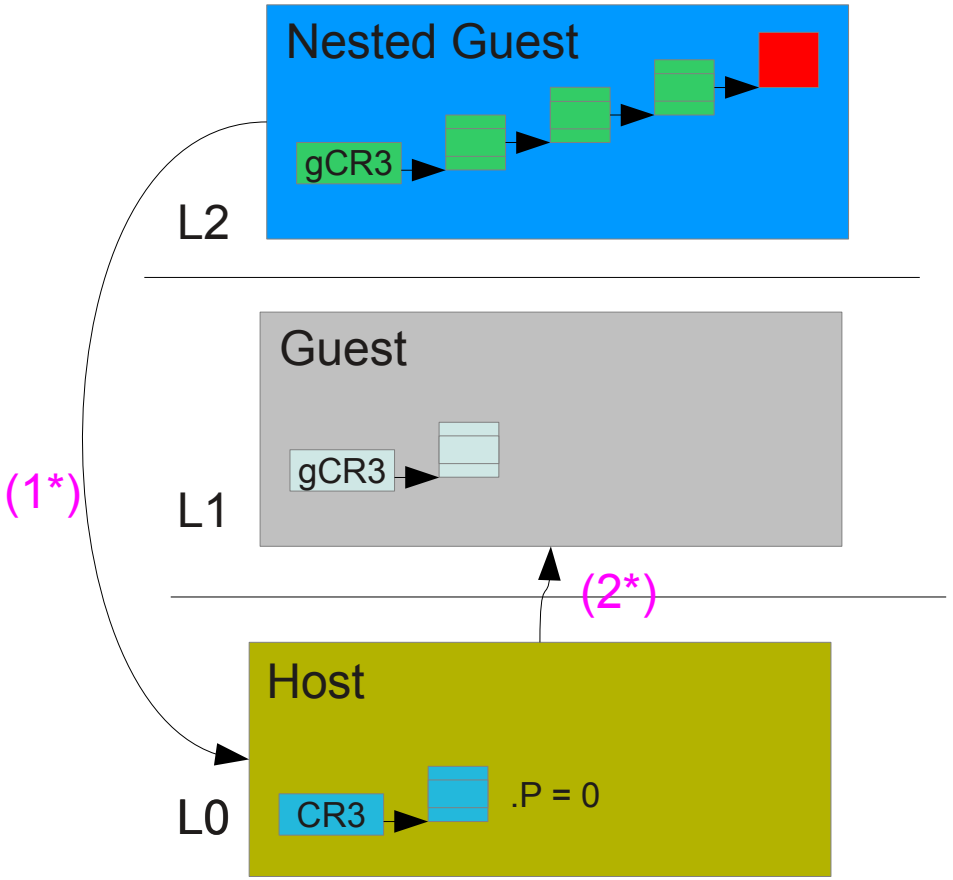

NGVA: nested guest virtual address
NGPA: nested guest physical address
GVA: Guest virtual address
GPA: Guest physical address
HVA: host virtual address
HPA: host physical address

# Soft MMU on Soft MMU

- It is the software only solution
- Host (L0) offer shadow page tables to translate NGVA to HPA
- Guest ( L1) offer shadow page tables to translate NGVA to GPA which need to be write-protected
- Need to intercept the MMU events of Nested guest (L2).

# Soft MMU on Soft MMU

- Nested guest memory access




At the beginning, both shadow page tables on host and guest are empty so:

(1*)

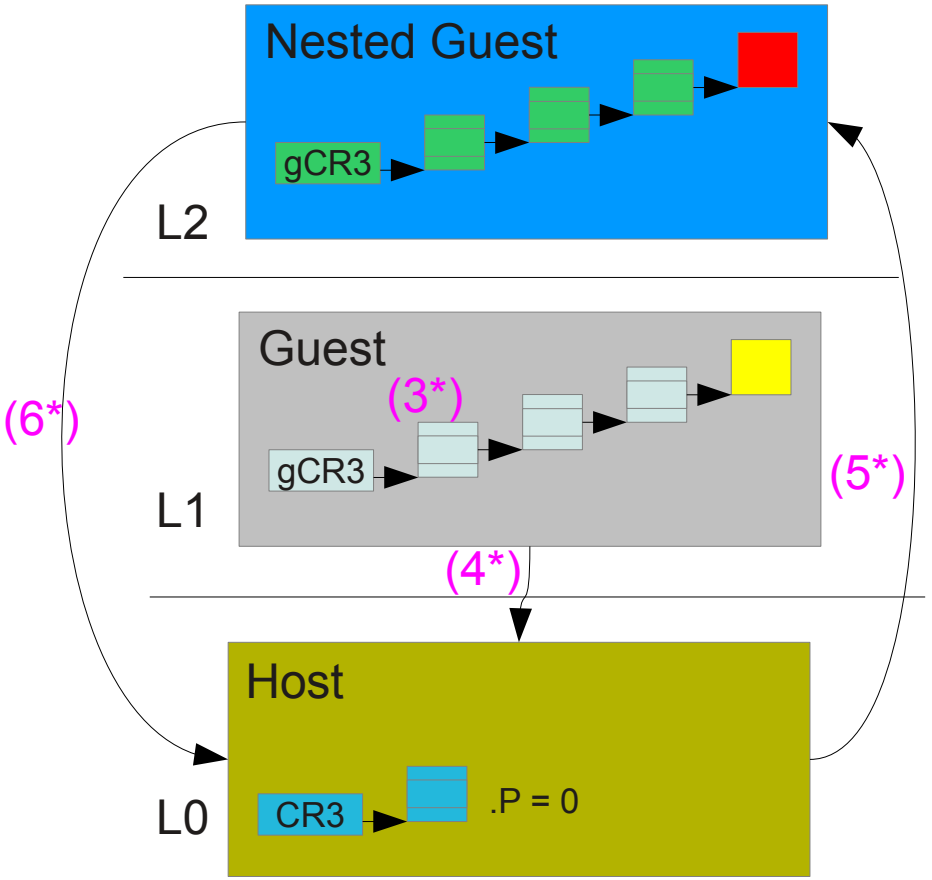
The nested guest accesses memory can generate #PF which can be intercepted by host.

(2*)

Walking guest's page table the host sees the mapping is invalid, it resumes guest (L1) and injects the #PF into guest (L1).

# Soft MMU on Soft MMU

- Nested guest memory access




(3*)

Guest receives the #PF, then fixes its shadow page table which is used to translate **NGVA to GPA**.

(4*)

Guest executes VMRESUME to resume Nested guest which generates VM-exit and causes guest exits to Host.

(5*)

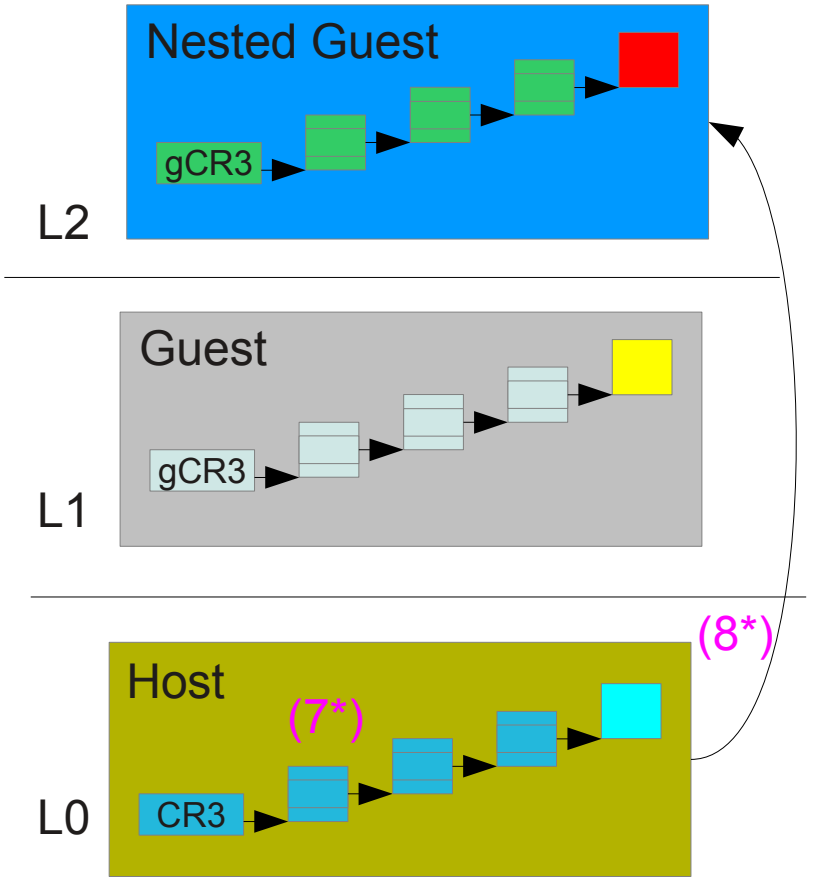
Host emulates VMRESUME which is called from guest, then return to Nested Guest.

(6*)

The nested guest re-executes the fault Instruction and cause #PF again.

# Soft MMU on Soft MMU

- Nested guest memory access




(7*)

Since the guest's shadow page table is valid, host can fix its shadow page table which maps **NGVA to HPA**.

(8*)

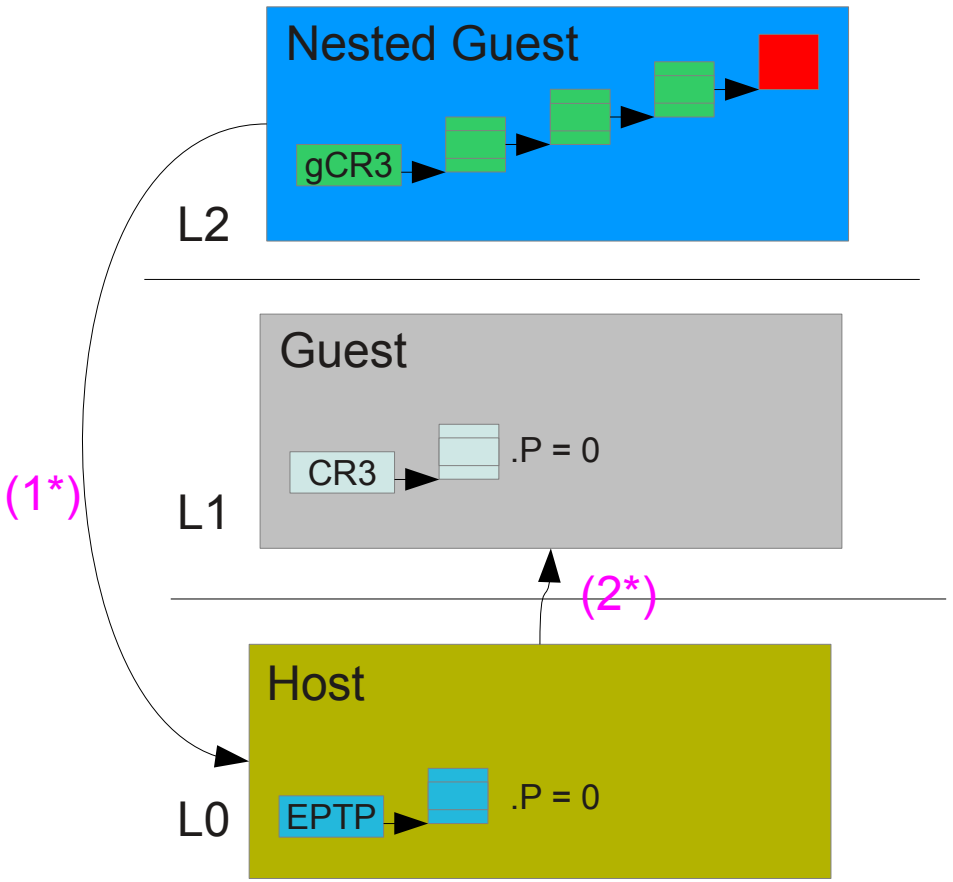
Resume nested guest. Since the mapping is valid now, it can happily access the memory.

# Soft MMU on Hard MMU

- Hard MMU is enabled on Host but disabled on Guest
- Host (L0) offer shadow pages which are loaded into VMCS.EPTP to translate NGPA to HPA
- Guest (L1) offer shadow pages to translate NGVA to GPA which are loaded into CR3 register and write-protected
- #PF and the MMU events in nested guest (L2) should be intercepted

# Soft MMU on Hard MMU

- Nested guest memory access




At the beginning, both shadow page table on guest and host are empty so:

(1*)

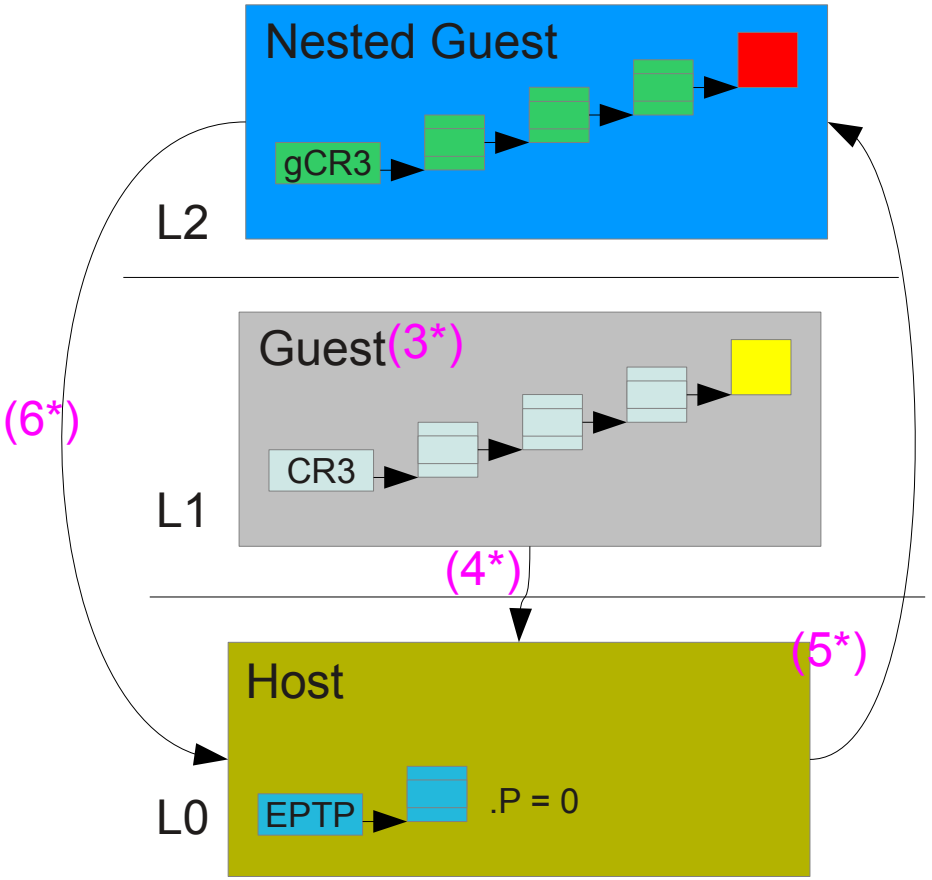
The nested guest accesses memory can generate #PF which can be intercepted by host.

(2*)

Directly inject #PF to guest (since #PF indicates the mapping is invalid when translate NGVA to GPA).

# Soft MMU on Hard MMU

- Nested guest memory access




(3*)

Guest receives the #PF, then fixes its shadow page table which is used to translate **NGVA to GPA**.

(4*)

Guest executes VMRESUME to resume nested guest which generates VM-exit and causes guest exits to Host.

(5*)

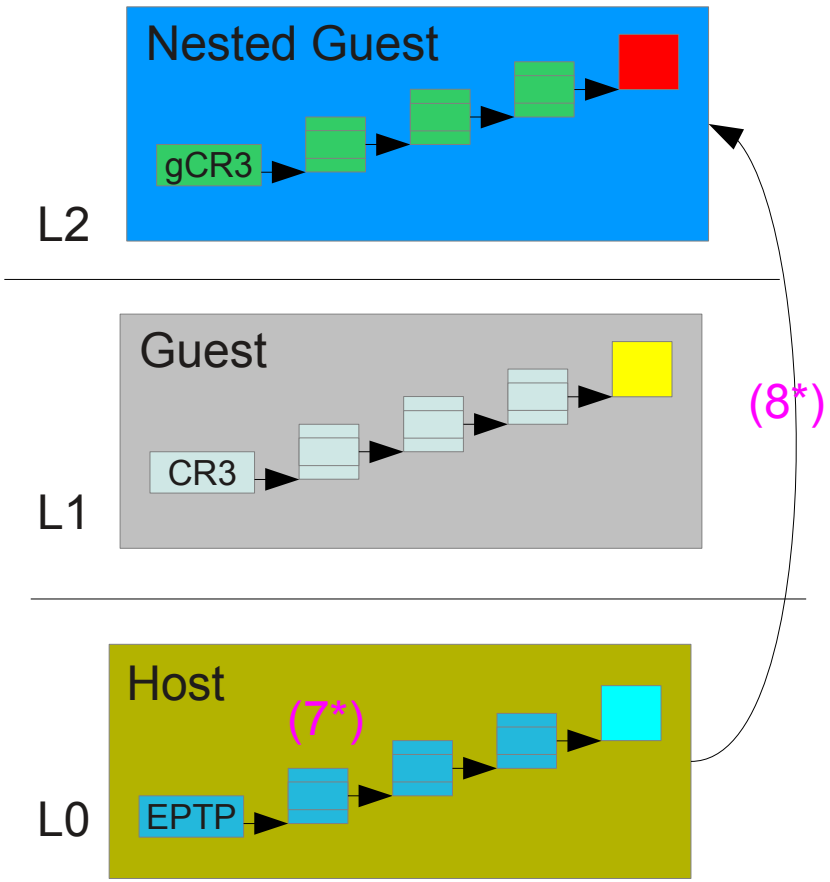
Host emulates VMRESUME which is called from guest, then return to Nested Guest.

(6*)

The nested guest re-executes the fault instruction and cause EPT MISCONFIG / EPT VIOLATION.

# Soft MMU on Hard MMU

- Nested guest memory access




(7*)

Host fixes its EPT page table which used to map **NGPA to HPA.**

(8*)

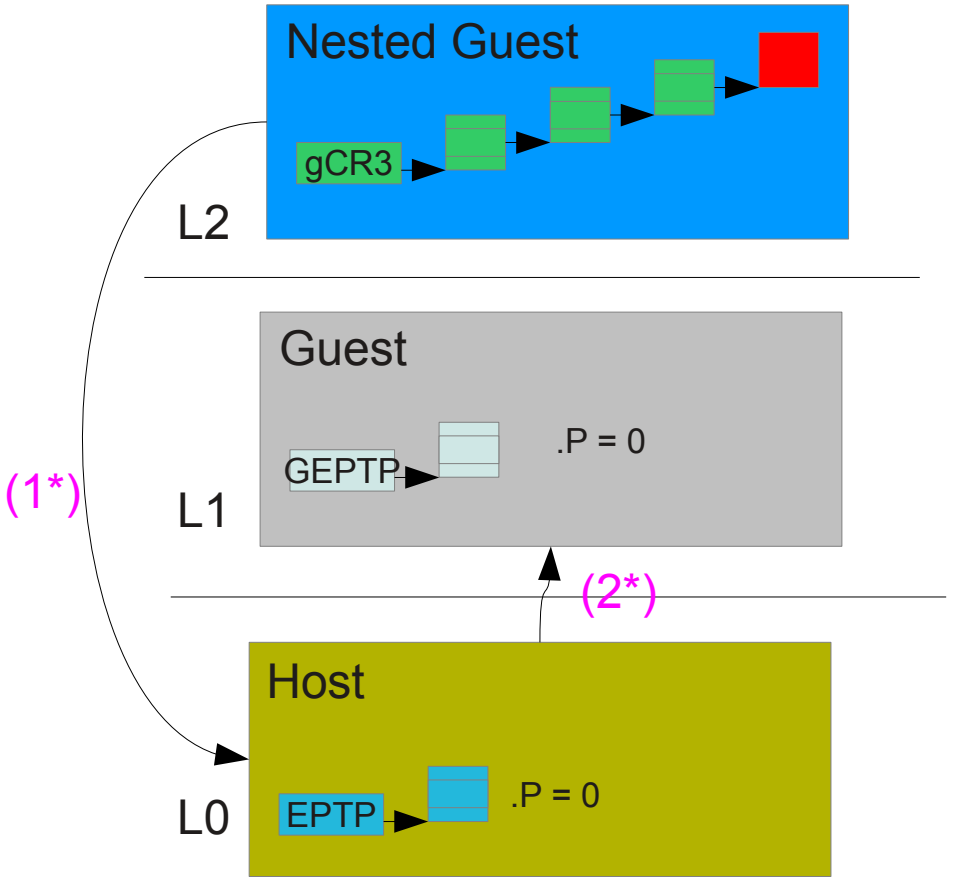
Resume nested guest since the mapping is valid now, it can happily access the memory.

# Hard MMU on Hard MMU

- Hard MMU is enabled on both host and guest
- It is very similar with Soft MMU – shadow Guest's EPT page table
- Host (L0) offer shadow pages to translate NGPA to HPA which are loaded into VMCS.EPTP
- Guest (L1) offer shadow pages to translate NGPA to GPA which should be write-protected
- #PF need not be intercepted.
- The EPT Page Table events in nested guest should be intercepted

# Hard MMU on Hard MMU

- Nested guest memory access




At the beginning, the EPT page tables on both guest and host are empty so:

(1*)

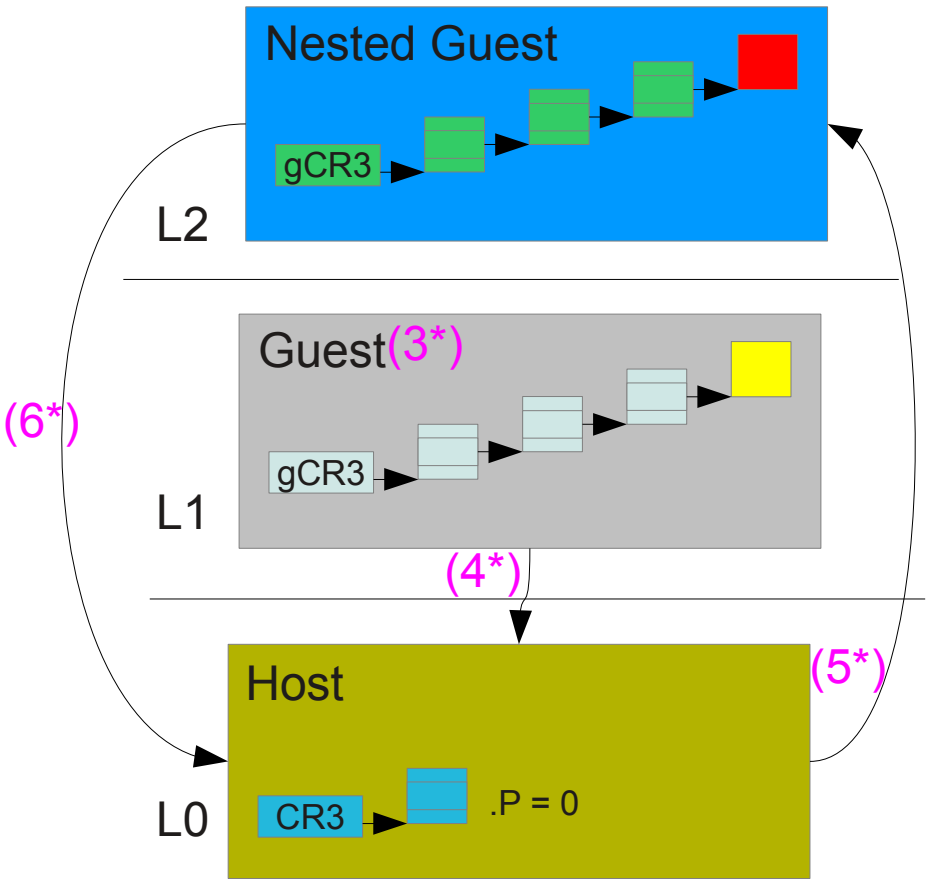
The nested guest accesses memory can generate EPT Misconfig/Violation which can be intercepted by host.

(2*)

Walking guest's EPT page table the host sees the mapping is invalid, it injects the EPT Misconfig/Violation into guest.

# Hard MMU on Hard MMU

- Nested guest memory access




(3*)

Guest receives the EPT Misconfig/Violation, then fixes its EPT page table which is used to translate **NGPA to GPA**.

(4*)

Guest executes VMRESUME to resume nested guest which generates VM-exit and causes guest exits to Host.

(5*)

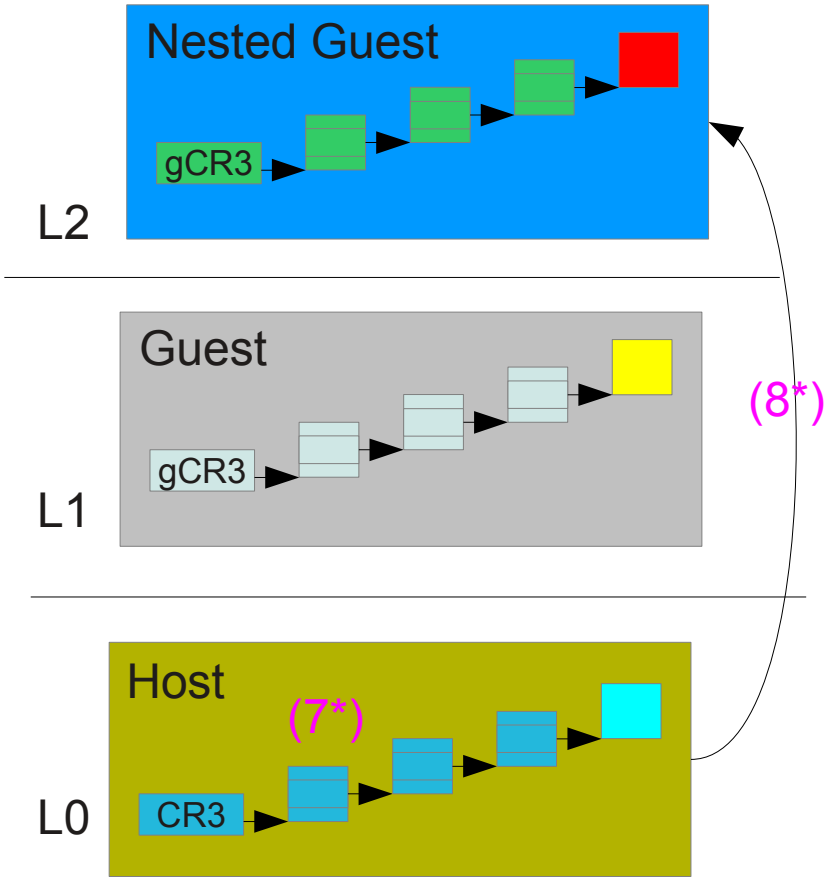
Host emulates VMRESUME which is called from guest, then return to Nested Guest.

(6*)

The nested guest re-executes the fault instruction and cause EPT Misconfig/Violation again.

# Hard MMU on Hard MMU

- Nested guest memory access




(7*)

Since the guest's EPT page table is valid, Host can fix its shadow page table which maps **NGPA to HPA**.

(8*)

Resume nested guest since the mapping is valid now, it can happily access the memory.

Questions?

# Reference

- kvm source code:
  - git://git.kernel.org/pub/scm/virt/kvm/kvm.git
- Documentation/virtual/kvm/nested-vmx.txt in kernel souce code
- http://www.usenix.org/events/osdi10/tech/full_papers/Ben-Yehuda.pdf

Thanks! :)