Hardware-Assisted Virtual IOMMU with Nested Translation

Author:Dr. Siqi Zhao
Affiliation: T-Head Semiconductor Co., Ltd.
Contact: zhaosiqi.zsq@alibaba-inc.com

Introduction:

To improve I/O performance, physical devices can be directly assigned to virtual machines with assistance of an IOMMU; a technique commonly known as 'device passthrough'. The passthrough devices can be directly accessed by the guest kernel with Guest Physical Address (GPA). However, existing IOMMU infrastructure provides limited support for virtual IOMMU.
The existing IOMMU infrastructure relies on two approaches to provide virtual IOMMU support to virtual machines. The first approach is emulation and the second approach is paravirtualization.

The emulated virtual IOMMU requires the host to track and combine the page table mappings configured by the guest. Such a process is lengthy and involved. The paravirtual IOMMU requires explicit collaboration between the guest and the host.

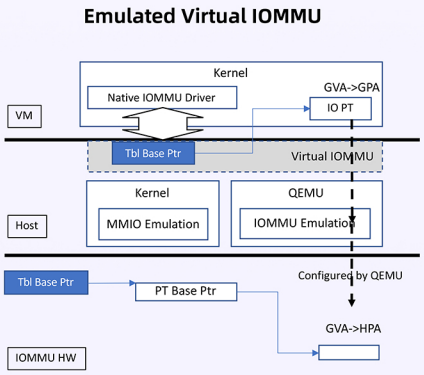

Figure 1: Emulated Virtual IOMMU

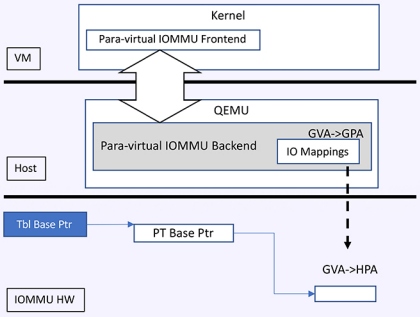

Figure 2:para-virtual IOMMU

Reference Design:

Hardware-Assisted Virtual IOMMU

The guest is presented with an IOMMU that is identical to the physical IOMMU, therefore, it can directly reuse the driver written for the host. Furthermore the guest's translation tables are directly used by the hardware IOMMU to translate DMA addresses, eliminating any need to synchronize and combine translation mappings.
The design uses a memory-resident region to store the register state of the virtual IOMMU. The data at a given offset within the region represents the value of the register of the virtual IOMMU at that offset within the hardware MMIO range. This page is mapped to the guest in the G-stage mappings, therefore, the guest's IOMMU driver can directly access the page. For example, the guest fills the GPA of the root of its translation tables into this page when the driver configures the root register. Figure 3 illustrates this architecture.
When the hardware IOMMU is configured to perform nested translation for a given device, the device's translation tables store a pointer to the aforementioned virtual IOMMU stage region along side the pointer to the G-stage page tables. After obtaining the pointer to the memory region when the hardware walks the translation tables, it follows the translation tables configured by the guest via the root pointer filled by the guest, treating all the addresses as GPA and translating them via walking the G-stage page tables. In the end, the GVA in the DMA address is translated to the HPA, and the transaction is forwarded to the fabric for memory access.

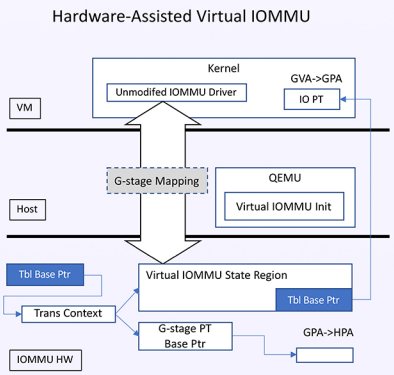

Figure 3: Hardware-Airtual IOMMU

Implementation Status:

We have implemented our design in QEMU and Linux KVM.

For more information