PASID Management in KVM

Yi Liu yi.l.liu@intel.com
Jacob Pan jacob.jun.pan@intel.com

Oct. 30th 2020

Disclaimers

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.

The products and services described may contain defects or errors known as errata which may cause deviations from published specifications. Current characterized errata are available on request. No product or component can be absolutely secure.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting www.intel.com/design/literature.htm.

Intel and the Intel logo are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others.

Agenda

- PASID Recap
- Usages
  - PASID in Shared Virtual Addressing (SVA)
  - PASID in Intel® Scalable IOV
- PASID Management

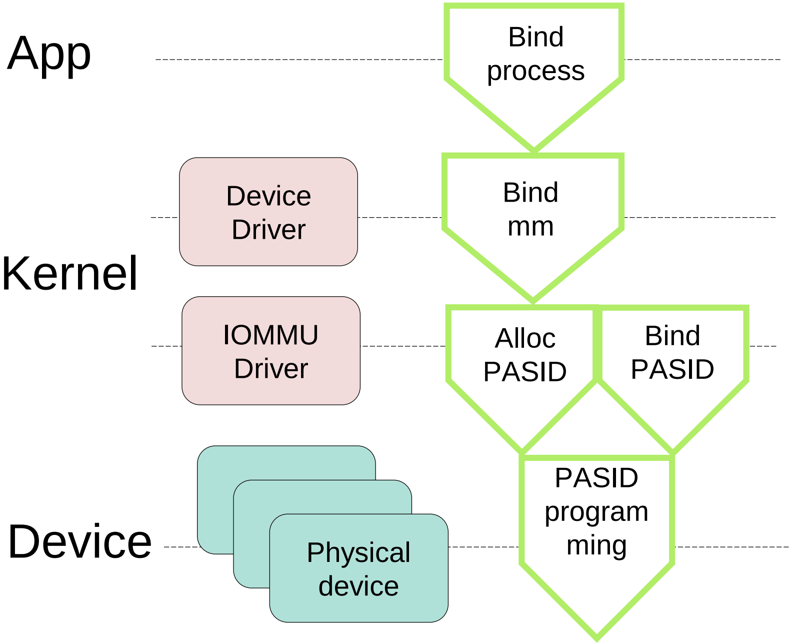
PASID Recap

- PASID (Process Address Space ID)
  - Defined by PCIe* spec
  - DMA remapping happens at RID & PASID granularity

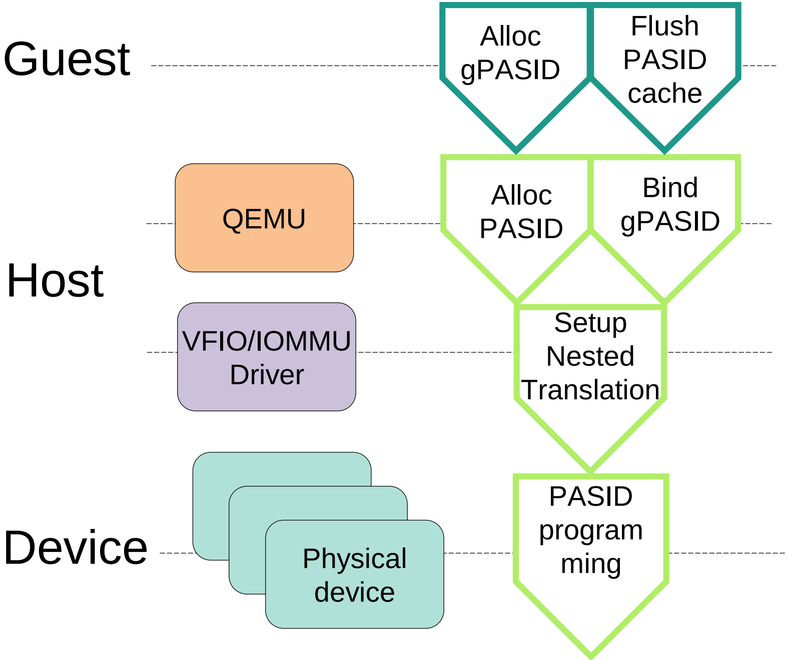
- IOMMU PASID Table
  - Per-device table by hardware design.
  - Storage in virtualization environment (Nested Translation[1])
    - Intel® VT-d: maintained by host
    - ARM* SMMUv3, AMD* IOMMU: maintained by guest

[1] https://static.sched.com/hosted_files/kvmforum2018/52/kvm-forum-vSVA-yliu-jpan-jean-eric.pdf

PASID in SVA



Native



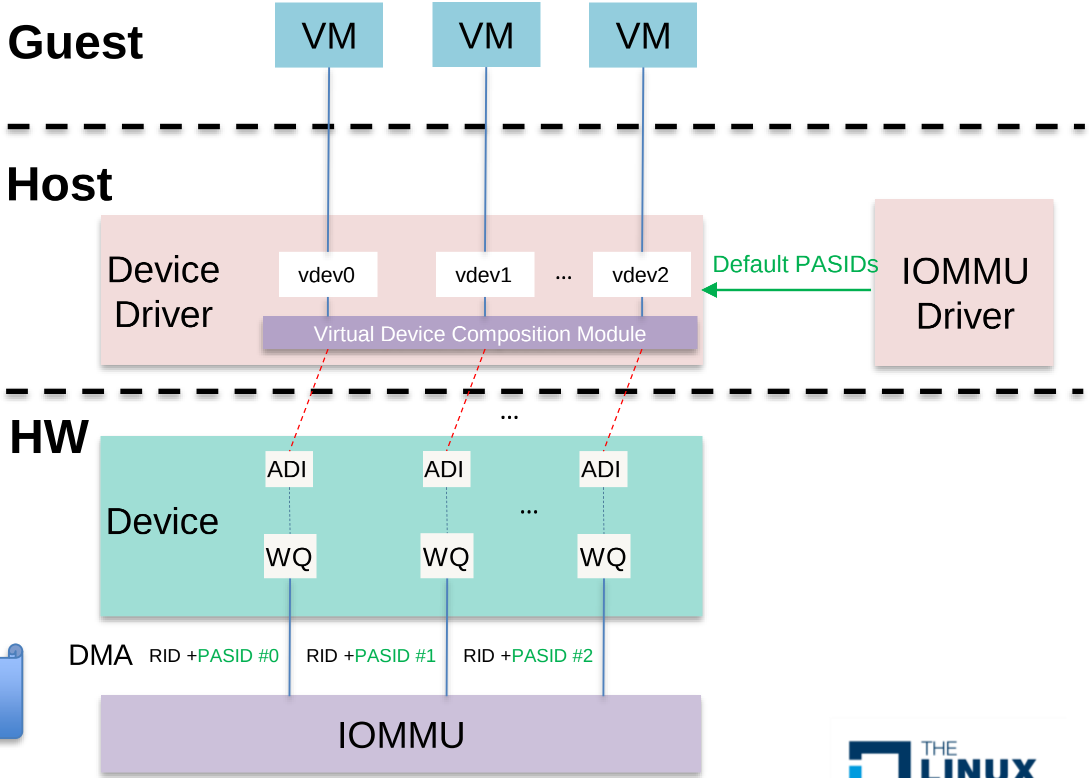
Virtualization (Intel® VT-d)

Could be binding guest PASID table per vendor support

PASID in Intel® Scalable IOV

- Default PASID of auxiliary domain
  - Each ADI (Assignable Device Interface[1]) has a default PASID
  - Assigned once attached to an auxiliary IOMMU domain[1]
  - ADIs attached to the same IOMMU domain share the default PASID of the domain
  - Programmed to hardware by parent device driver

PASID is I/O Address Space ID

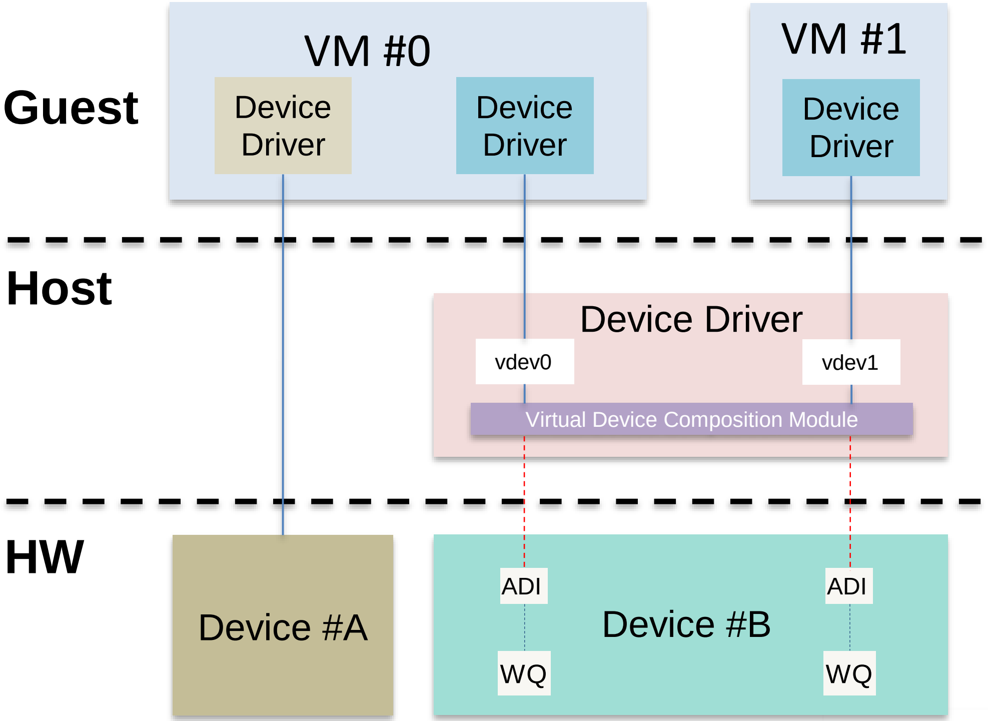


[1] https://events19.linuxfoundation.org/wp-content/uploads/2017/12/Hardware-Assisted-Mediated-Pass-Through-with-VFIO-Kevin-Tian-Intel.pdf

SVA & Scalable IOV

- PASID programming for PF
  - PASIDs from guest are programmed to hardware directly

- PASID programming for ADI
  - PASIDs from guest are converted to host PASIDs and then programmed to hardware
  - Intel® ENQCMD instruction can do PASID Translation in hardware
    - [“Scalable Work Submission in Device Virtualization - Hao Wu, Intel”]



PASID Programming in SVA usage

PASID Management w/ IOASID Core

IOASID is a generic kernel library (since v5.5) for managing PASIDs

- Guest-Host PASID Mapping
- Partitioning & Namespaces
- Synchronization/Notifications
- Lifecycles

Guest-Host PASID Mapping

1. Shadow guest PASID table (Intel VT-d SM®)

- Requires G-H PASID translation (H-PASID != G-PASID)
- Requires host PASID backing of each guest PASID
- Requires system-wide host PASID namespace due to shared workqueues (SWQ) (i.e. a single SWQ assigned to two VMs, the backing host PASIDs must be unique)
- PASID programming on PF assignment is NOT mediated, guest PASID is programmed. Potential conflict with ADIs on the same VM if guest PASID bind and PRS not enforced.

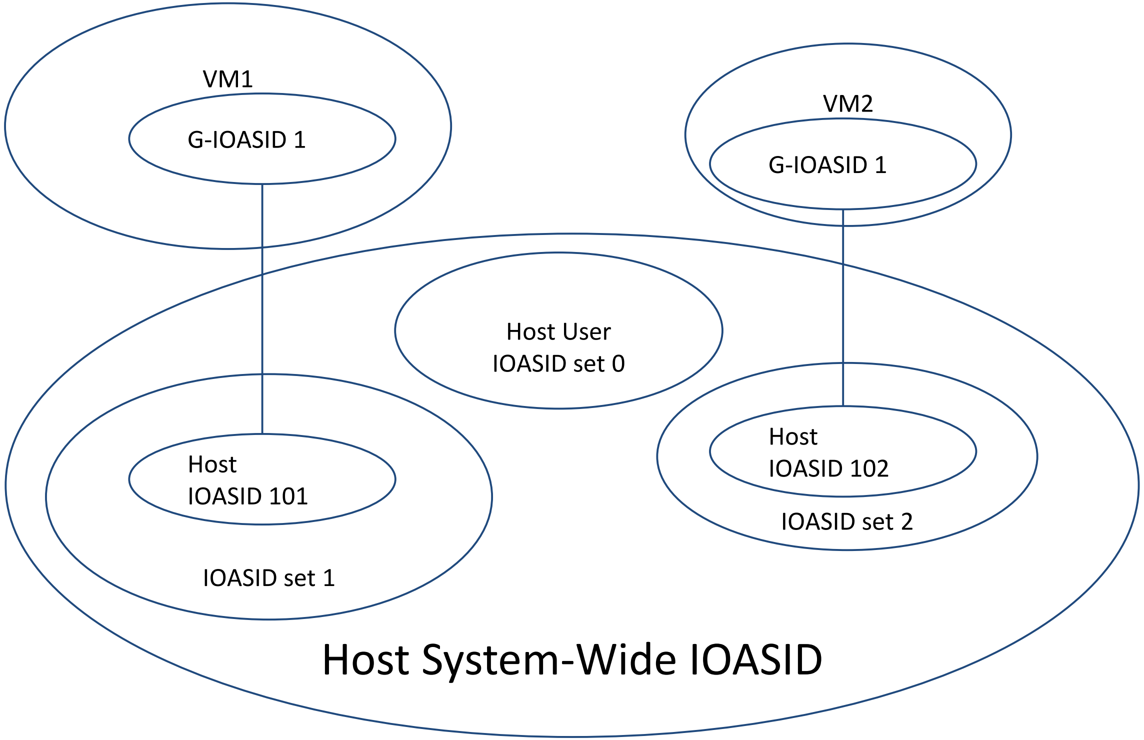
2. Bind guest PASID table (ARM SMMU® v3)

- PASID namespace can be per VM since host IOMMU walks guest PASID table
- Host doesn't care about guest PASIDs

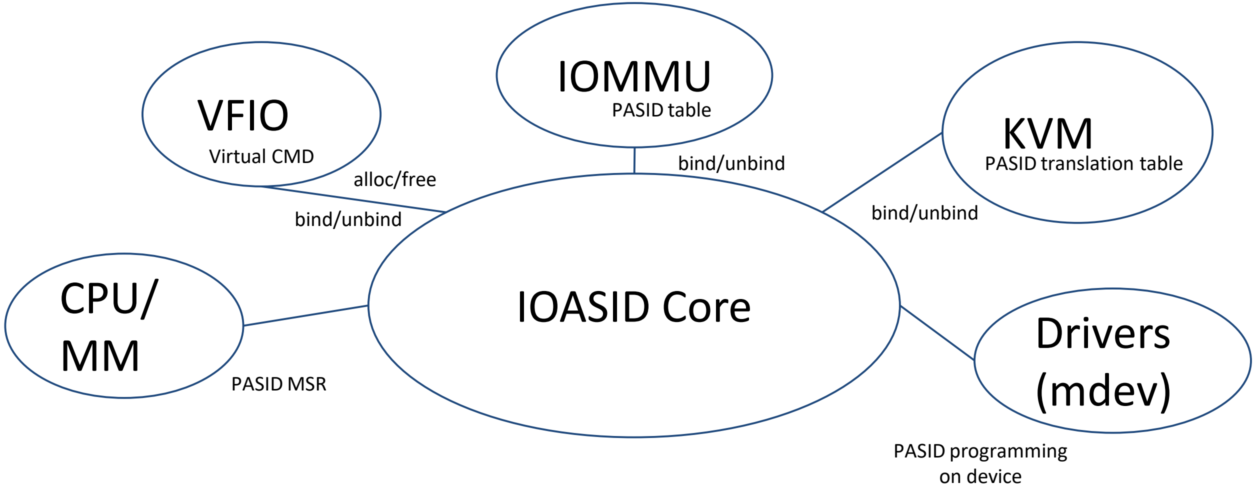
Requirements for #1 is a superset of #2!

Namespaces & Partitioning

1. A VM has its own PASID namespace
2. Host PASIDs are in a single namespace but partitioned into group/ioasid_sets

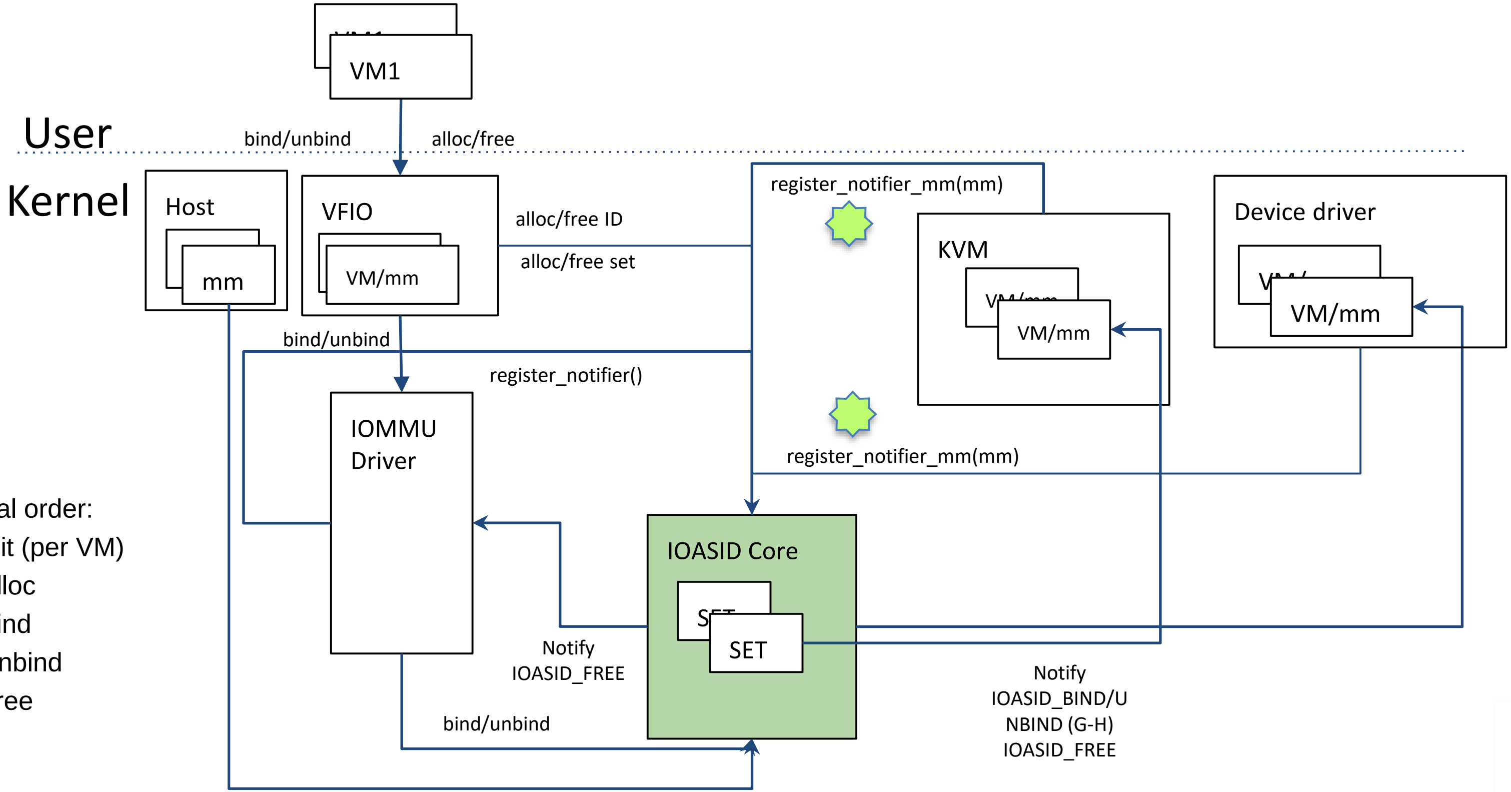


Synchronization among IOASID Use



https://software.intel.com/content/www/us/en/develop/download/intel-scalable-io-virtualization-technical-specification.html
https://software.intel.com/content/www/us/en/develop/download/intel-data-streaming-accelerator-preliminary-architecture-specification.html

Lifecycles: A PASID’s life



Normal order:

1. ✸ Init (per VM)
2. Alloc
3. Bind
4. Unbind
5. Free

Status

- Opens
  - Should we expose IOAPID allocation via VFIO or a new standalone UAPI?
  - How can user manage IOASID accounting/quota? Rlimit and Cgroup seem too heavy
- Patchsets
  - https://lkml.org/lkml/2020/9/28/1186
  - https://lore.kernel.org/linux-iommu/5dd95fbf-054c-3bbc-e76b-2d5636214ff2@redhat.com/T/

Summary

- DMA remapping is at RID & PASID granularity
- PASID is managed as I/O Address Space ID (IOASID) in Linux
- System-wide host PASID is chosen to support requirements from all vendors
- Guest has its own PASID namespace
- PASIDs can have multiple users with hardware context association
- Notifications and reference counting are used to manage lifecycles

KVM FORUM