Hardware-Assisted Mediated Pass-Through with VFIO

Kevin Tian
Principal Engineer, Intel

intel Software

Intel OpenSource TECHNOLOGY CENTER

Legal Disclaimer

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.

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.

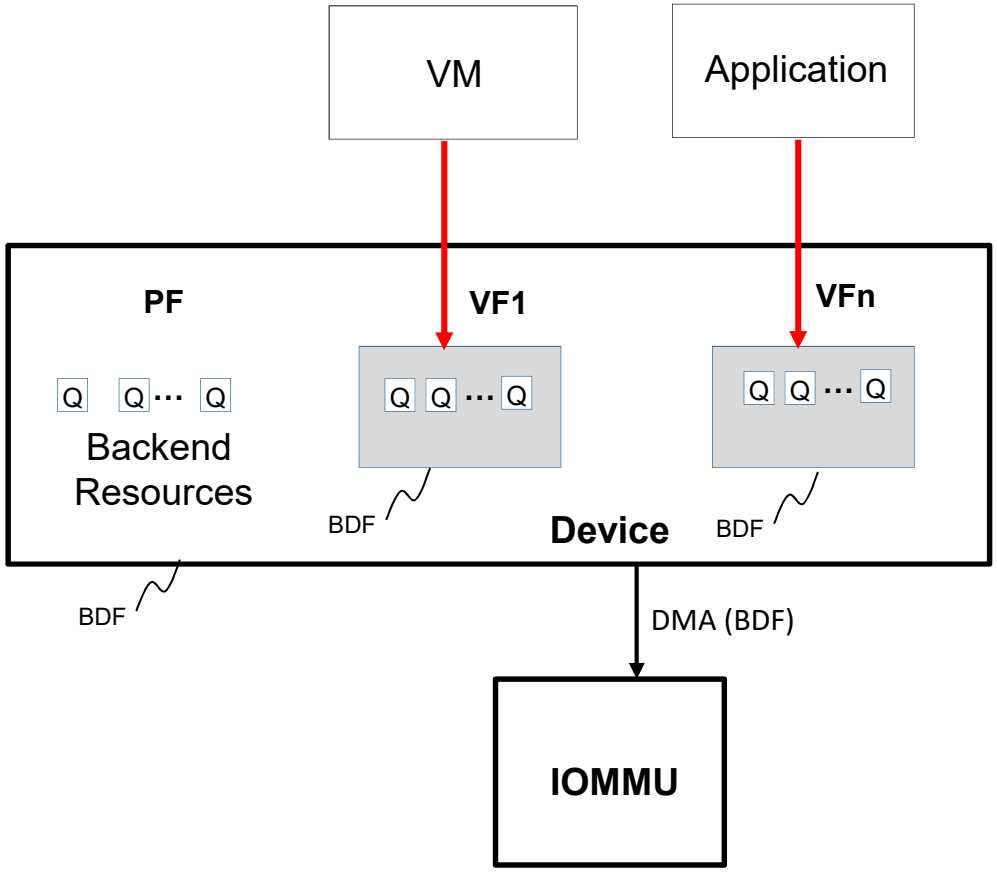
Intel and the Intel logo are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries.

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

VFIO

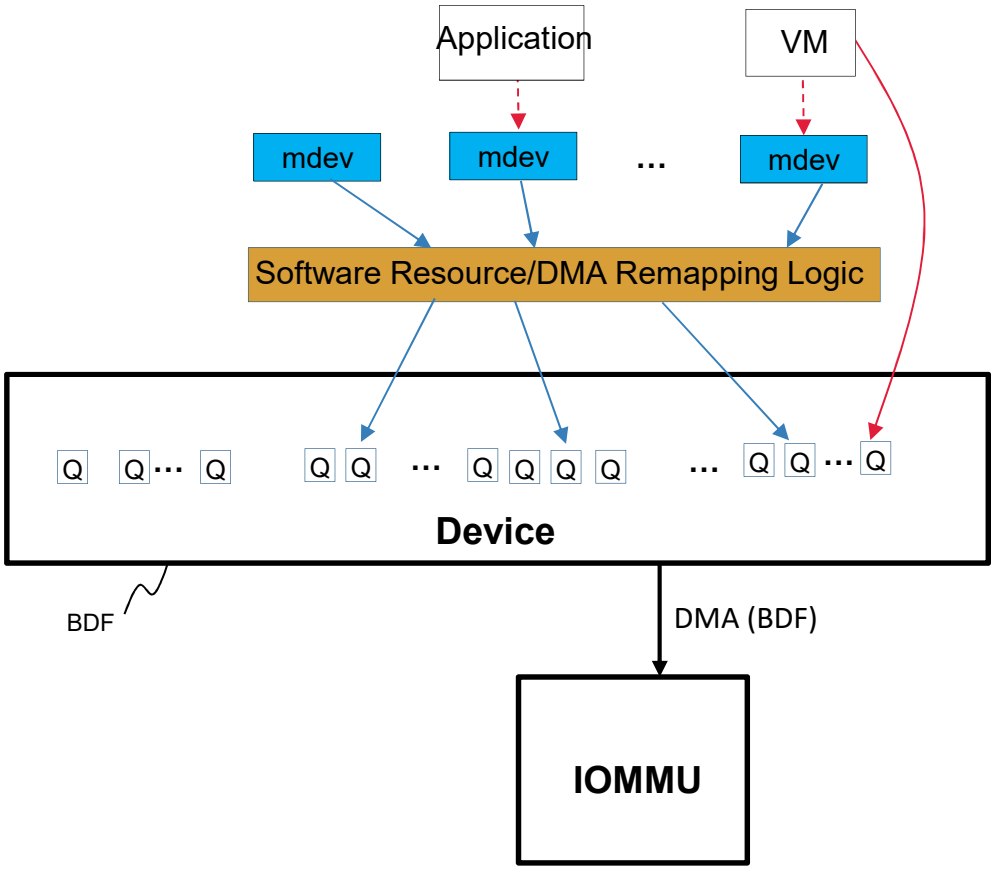
- A secure, userspace driver framework
- VFIO physical device
  - PCI endpoints, platform devices, etc.
  - PCI device sharing through PCIe® Single Root I/O Virtualization (SR-IOV)
- VFIO mediated device
  - vGPUs, channel I/O devices, crypto APs, etc.
  - Device sharing through vendor specific resource mediation policy

PCIe® SR-IOV



- Hardware-assisted I/O virtualization
  - Physical Function (PF)
  - Virtual Function (VF)
- Pros
  - Software simplicity
  - IOMMU-based DMA isolation
- Cons
  - Limited scalability
  - Fixed resource allocation
  - Lack of composability

Mediated Device

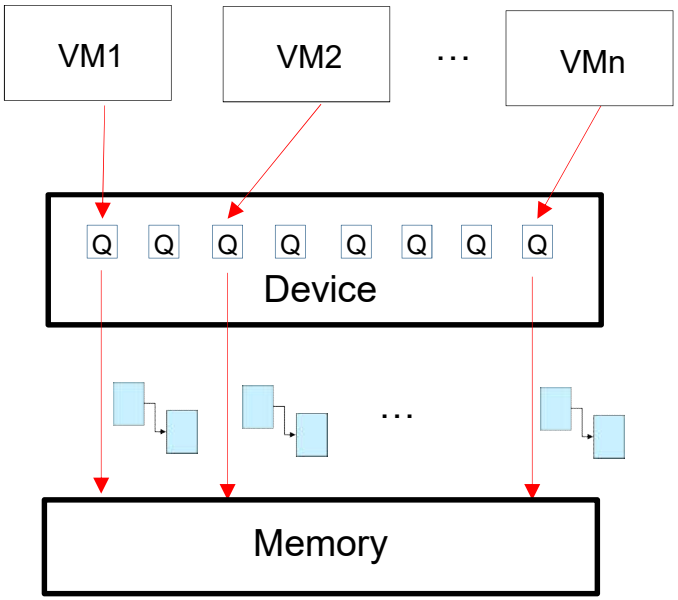


- Mediated pass-through architecture
  - Slow-path operations emulated by software
  - Fast-path operations passed through
- Pros
  - Flexible resource allocation
  - Composability
- Cons
  - Software-based DMA isolation (thus increases complexity and limits scalability)

Can we enjoy merits from both sides for hyper-scaled usage?

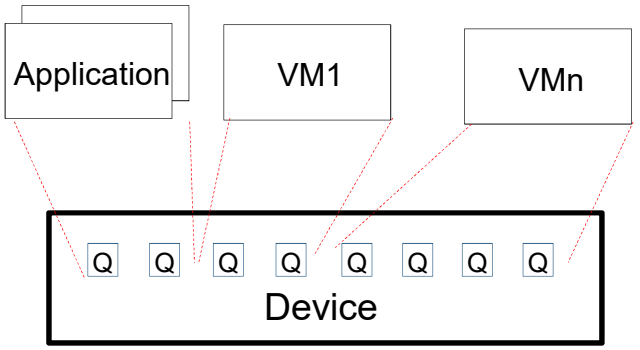
Goals

Finer-grained resource sharing

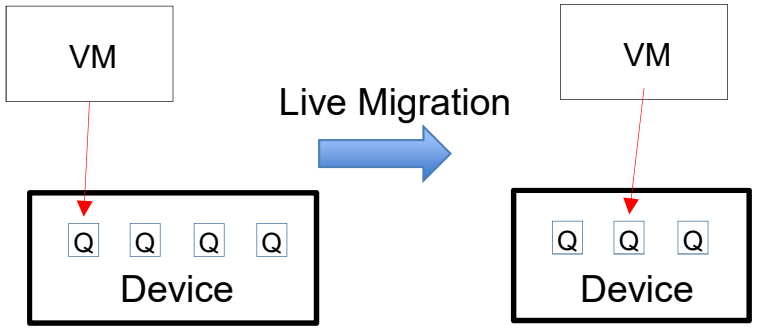


Hardware-enforced DMA Isolation

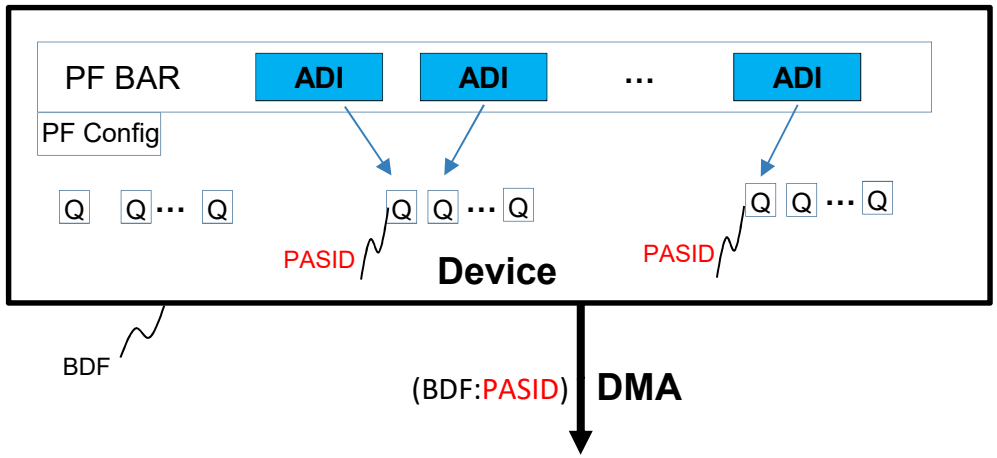
Flexible Resource Allocation



Composability



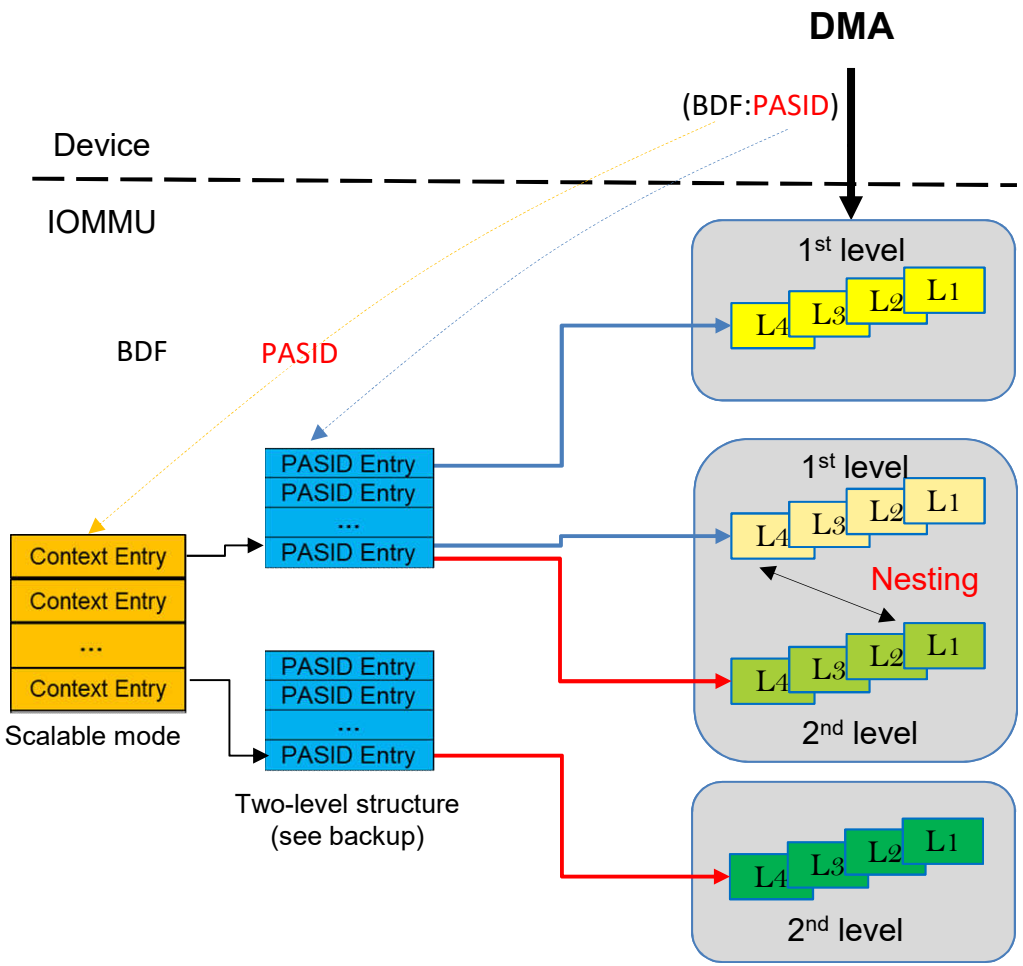
Assignable Device Interfaces (ADI)



- ADI - represents minimal sharable resource
  - Queues, queue pairs, contexts
  - Enumerated through DVSEC capability (see backup)
- Meets isolation criteria to be ‘assignable’
  - Functional isolation between ADIs
  - ADI MMIO in separate system page size regions
  - Independently resettable
  - Interrupt Message Storage (IMS)
  - …
- Tags DMA with unique PASIDs

Enable maximum possible scalability!

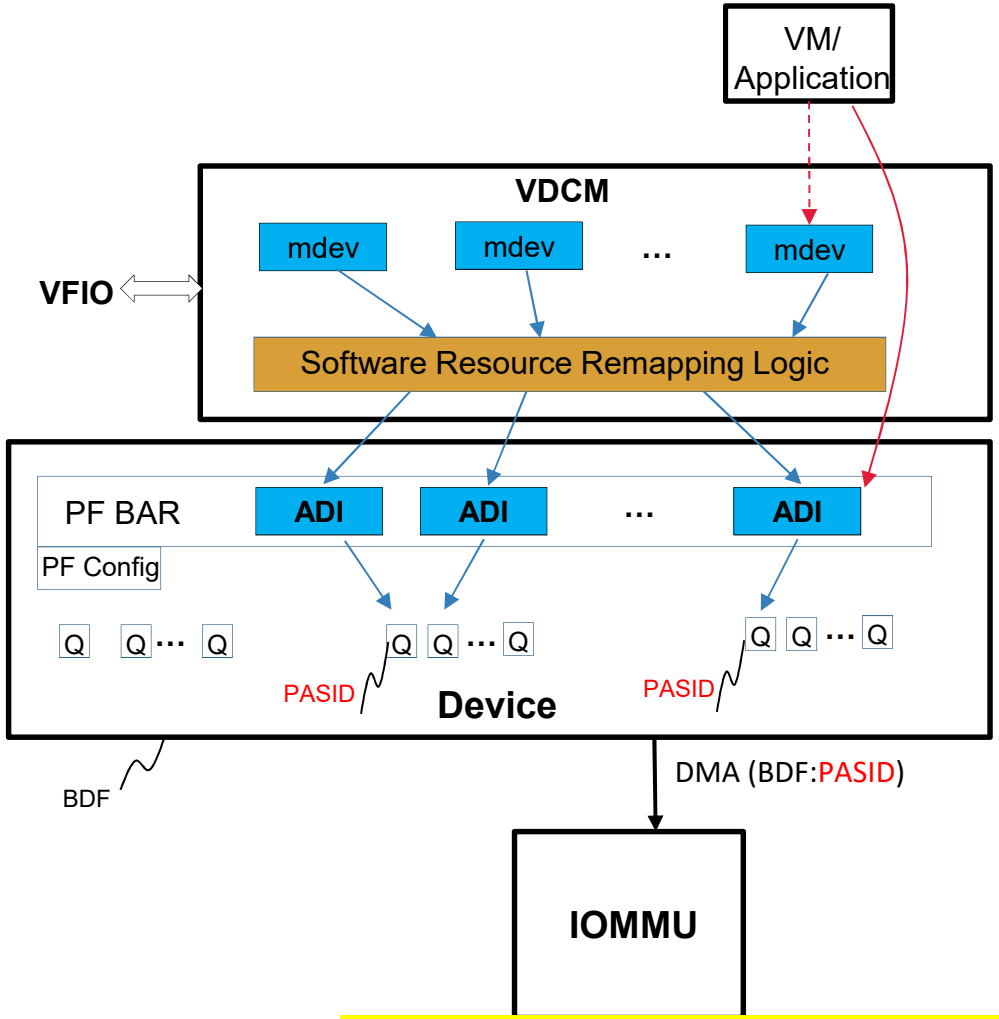
PASID-granular DMA Isolation



- Moves all IOMMU paging structures to per-PASID
  - 1st level translation
  - 2nd level translation
  - Nested translation
  - Pass-through translation
- Enables PASID-granular DMA isolation
- Supports all existing address translation usages
  - IOVA, VA, GPA, GIOVA and GVA

Enable hardware-enforced DMA isolation!

Software Composition



- Virtual Device Composition Module (VDCM)
  - Software managed resource remapping between mdev and ADI
  - Composes ADIs into mediated device (mdev)
- Leverage VFIO mdev framework for
  - Managing mdev life-cycle
  - Setting up access policy on mdev resources
  - Serving slow-path operations from guest

Enables great flexibility and composability!

Combining them together…

Intel® Scalable I/O Virtualization (Intel® Scalable IOV)

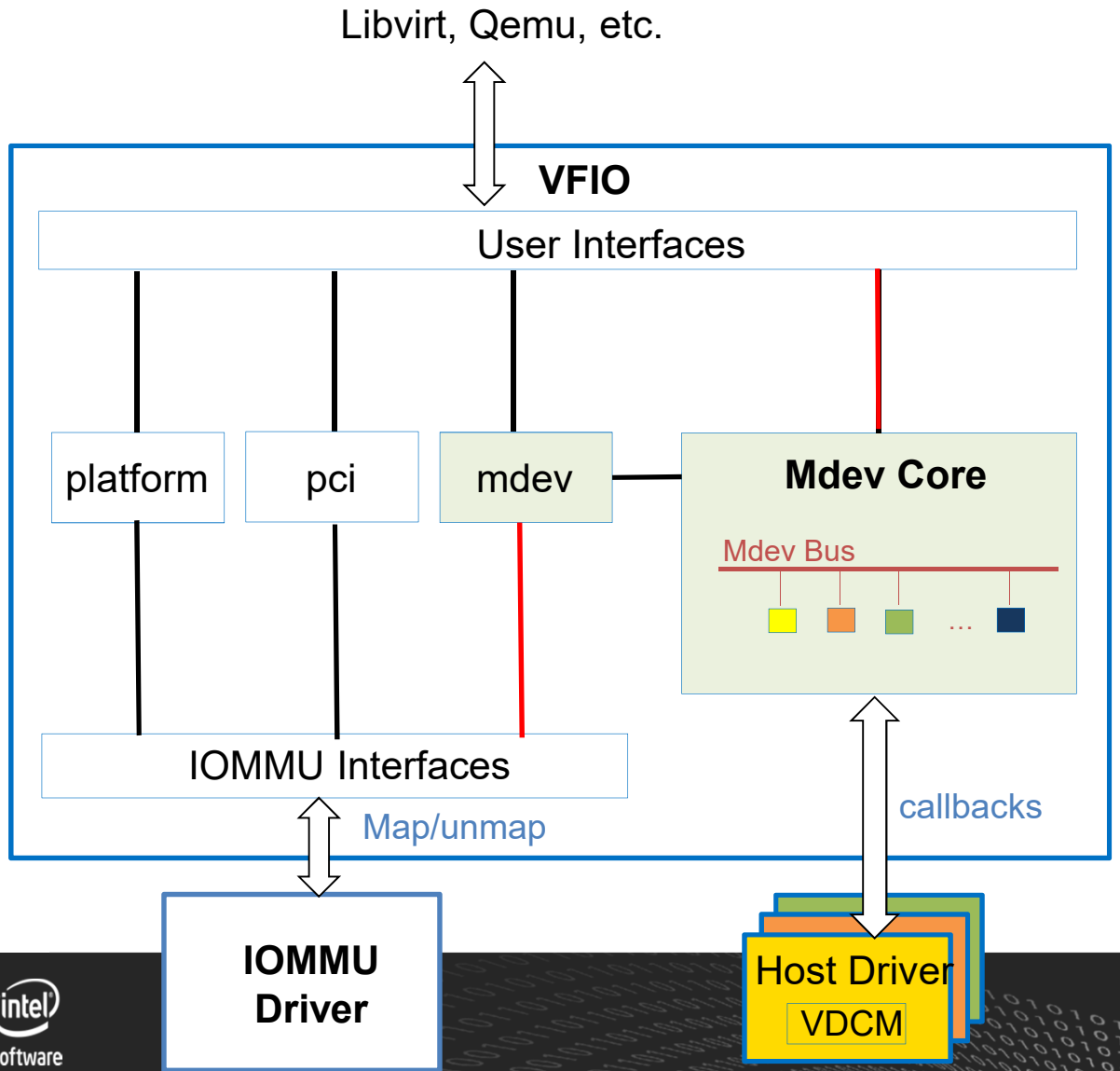
- A hardware-assisted mediated pass-through architecture
  - Device: supports Assignable Device Interfaces (ADIs)
  - Platform: extends Intel® VT-d with PASID-granular DMA isolation (*scalable mode*)
  - Software: moves infrequent (slow-path) accesses from device to software
- Supports any type of devices
  - e.g. NIC, storage, GPU, accelerators, … (integrated or discrete)
- Supports both VM and bare-metal usages

Documentation

- Intel® VT-d specification update (Rev 3.0)
  - Documents Intel® VT-d (IOMMU) support for PASID granular address translation
- Intel® Scalable I/O Virtualization Technical Specification (Rev 1.0)
  - Documents the Scalable IOV architecture blueprint and operation, including DVSEC
  - Addresses architecture requirements for devices and drivers
  - Agnostic of type of device or specific implementation
  - Openly published to enable broad device and software ecosystem
- https://software.intel.com/en-us/articles/intel-sdm

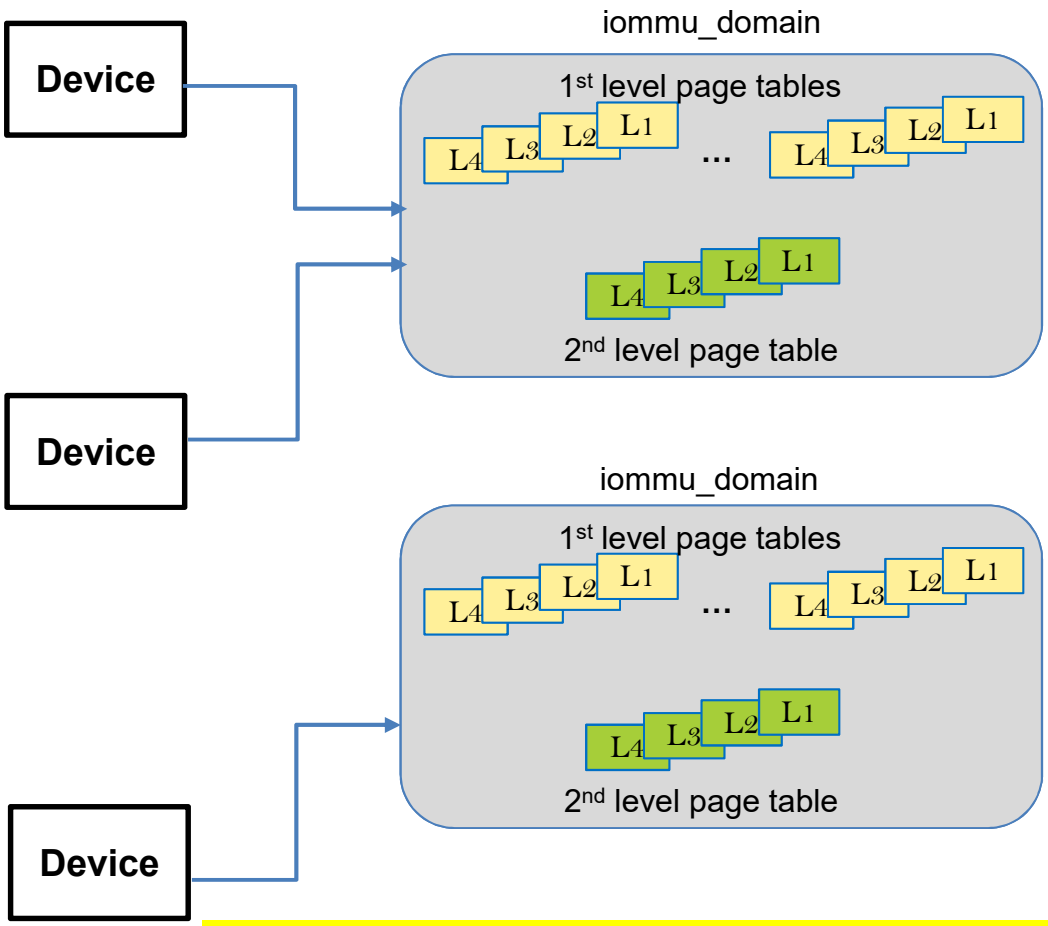
What does it mean to VFIO/IOMMU driver?

VFIO: IOMMU-capable Mdev



- IOMMU-capable mdev
  - Allow IOMMU operations on mdev
  - Opt-in by VDCM
- Finer-grained resource management
  - New aggregated type to compose any number of ADIs into a mdev

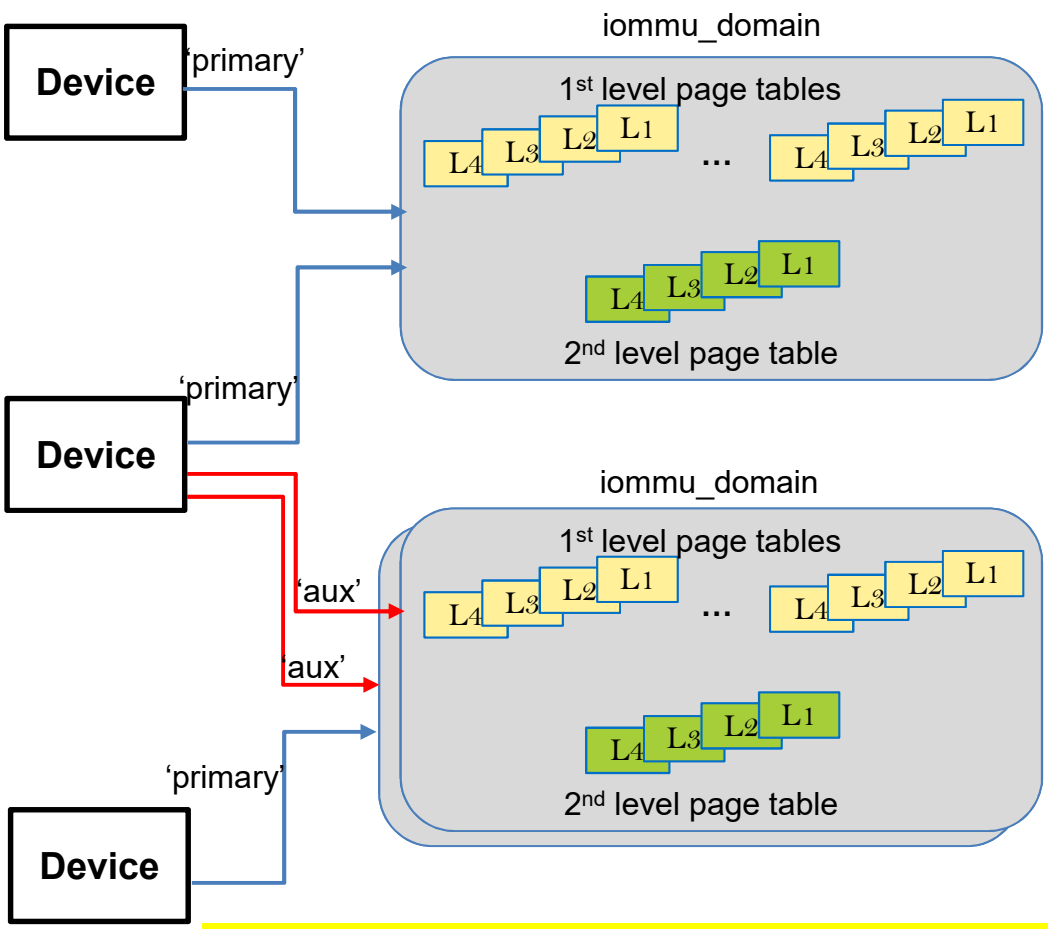
IOMMU Domains



- One device can attach to one domain, at any time
  - One domain can be attached to multiple devices
- Domain type describes IOMMU policy for device DMAs
  - DMA, UNMANAGED, IDENTITY, and BLOCKED
- Domain is switched when policy changes

"one domain" scheme doesn't work for IOMMU-capable mdev!

Auxiliary Domain



- One device can attach to multiple domains
  - A primary domain used for DMA-API
  - Multiple auxillary domains used for mdev instances
- 'aux' is a device attribute instead of domain attribute
  - Same domain may represent as 'primary' to deviceA and 'aux' to deviceB
  - 'primary' vs. 'aux' is decided at domain attach time
  - Device driver enables 'aux' capability on device before attaching domain
- No change to at(de)tach API
  - VFIO attaches domain to mdev's parent

Avoid mdev awareness in IOMMU layer!

Mdev with vIOMMU

- Ongoing effort to enable VFIO devices with vIOMMU
  - Shadow vIOMMU 2nd level usages (e.g. GIOVA)
  - Nesting vIOMMU 1st level usages (e.g. GIOVA/GVA)
    - Including system-wide PASID management
  - Cache invalidation forwarding (when nesting is in-use)
  - Page request/response handling (for guest SVA)
- Expect common user-space logic for vfio-pci and vfio-mdev
  - Just granularity difference handled within IOMMU driver
- Qemu: emulating new VT-d scalable mode emulation
- For more detail, join below session by Yi & Jacob!
  - "Shared Virtual Addressing in KVM"

Status

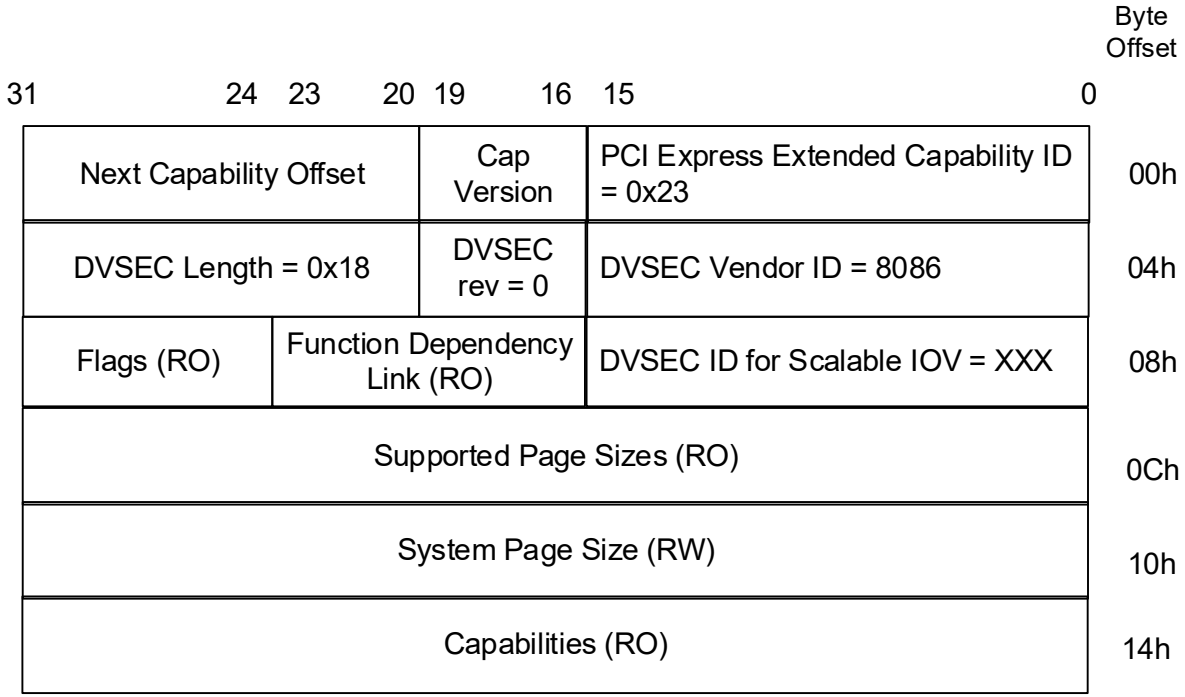
- Key developers
  - Baolu (Allen) Lu (baolu.lu@intel.com)
  - Yi Liu (yi.l.liu@intel.com)
  - Jacob Pan (jacob.jun.pan@intel.com)
- RFC patch progress
  - https://lkml.org/lkml/2018/10/7/54 for scalable mode support in intel-iommu driver in v3
  - https://lkml.org/lkml/2018/10/12/225 for aux_domain and IOMMU capable mdev in VFIO/IOMMU driver in v3
  - Continued hot discussion around vIOMMU/vSVM (in multiple threads)
    - Mdev requirement is being considered gradually
  - https://www.mail-archive.com/libvir-list@redhat.com/msg173811.html for aggregated mdev type in v3

Q/A

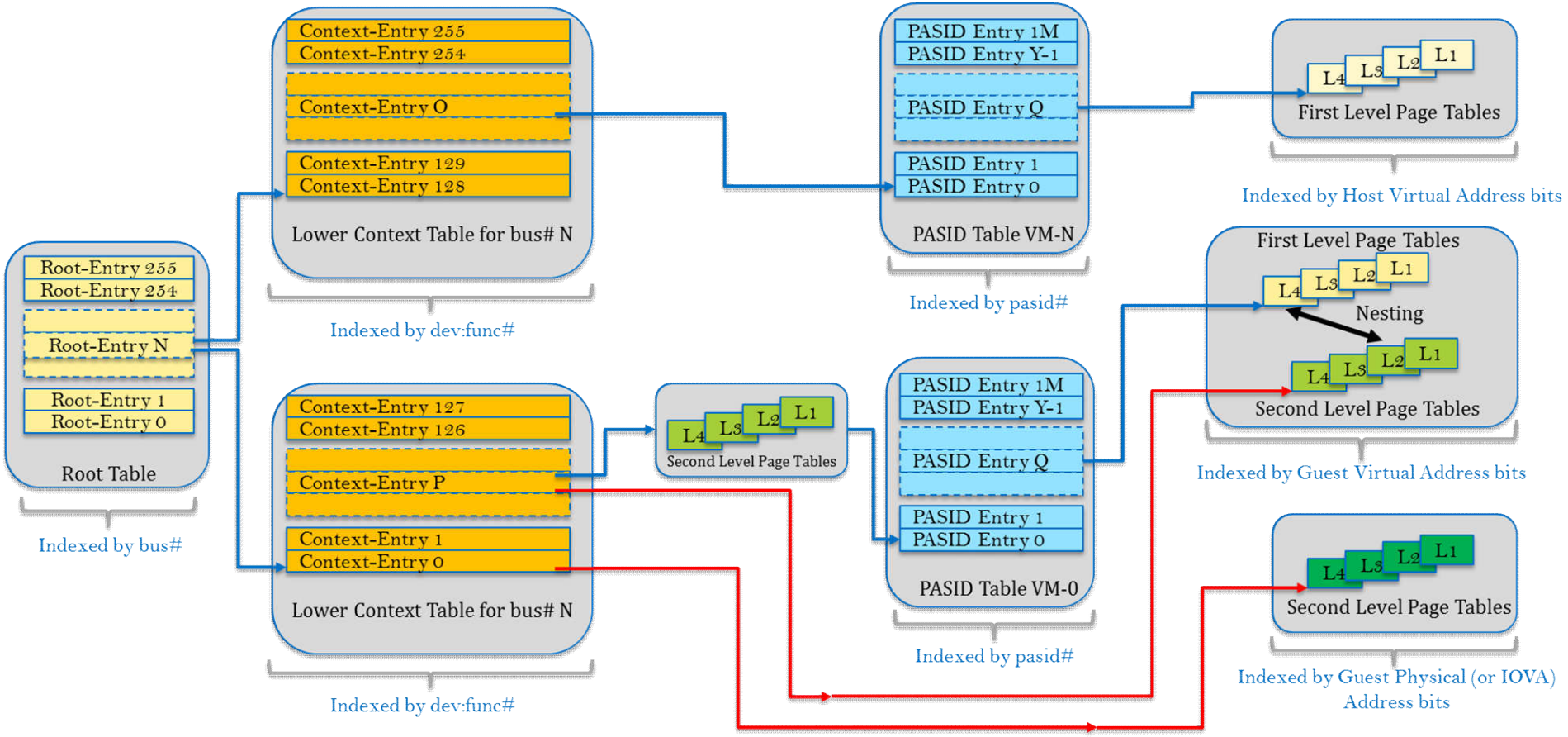
Backup

Enumeration of Intel® Scalable IOV Capability

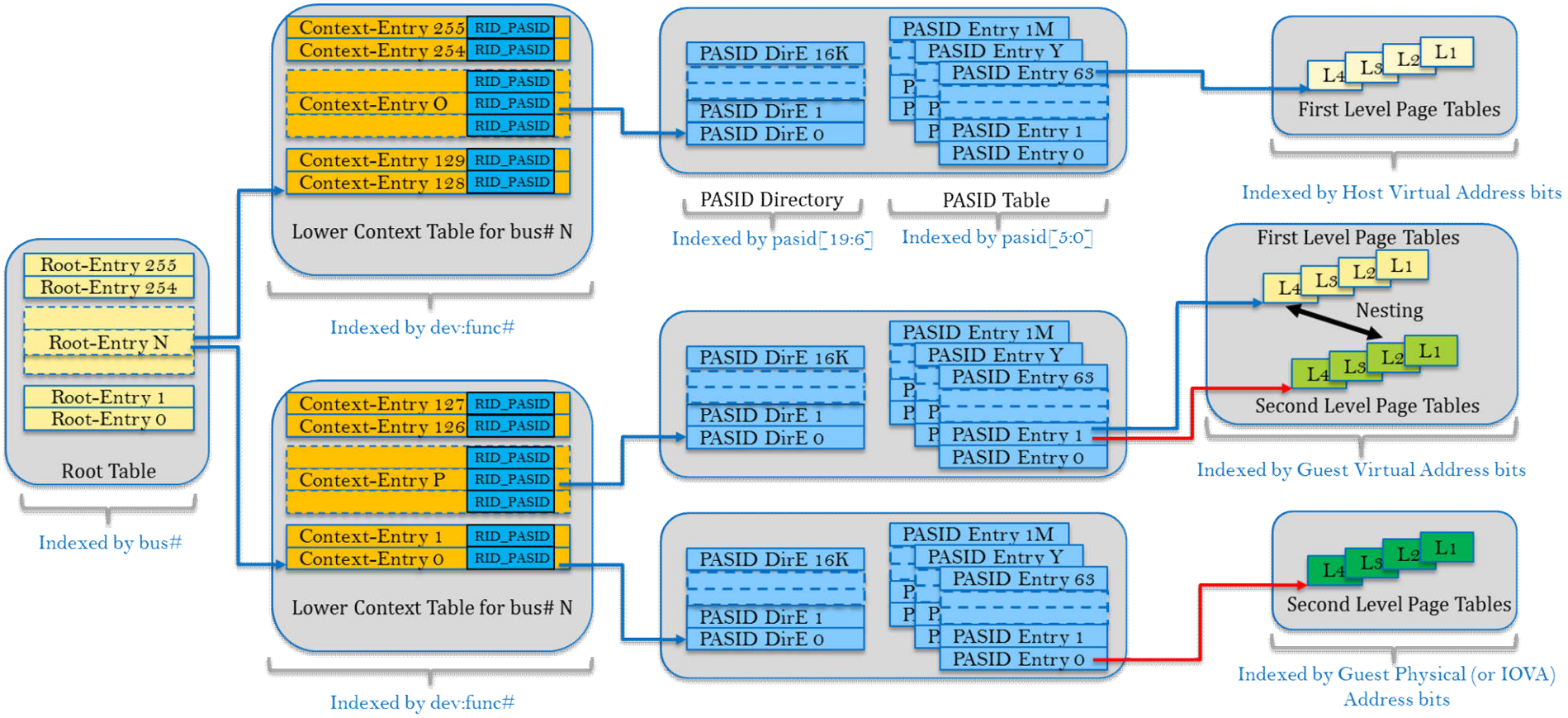
- Designated Vendor Specific Extended Capability (DVSEC) to discover Intel® Scalable IOV capability
  - A simplified subset of SR-IOV capability



VT-d Extended Context Mode (Deprecated)



VT-d Scalable Mode (New)



Key Difference: PASID is a global ID space shared by all VMs.

ALL page-table pointers moved to PASID Granular table