



Live Update IOMMU Roadmap and Arm SMMUv3

Samiullah Khawaja and Pranjal Shrivastava - Google



LINUX
PLUMBERS
CONFERENCE

PRAGUE, CZECHIA / OCT. 5-7, 2026

Liveupdate IOMMU Preserve/Restore Pattern

VFIO/IOMMUFD

IOMMU Core

IOMMU Driver

Preserve device and domain

Preserve domain,
device and
iommu
association

Preserve page tables,
root tables, pasid
directory and tables

Skip disable on
shutdown

Reboot

Next Kernel

Restored IOMMU
HW and domains

Reassociate domains with preserved devices.
The restored devices are locked and cannot be
used by any other driver.

Reclaim device and domain ownership

Transfer ownership and unlock domain
and devices

LINUX
PLUMBERS
CONFERENCE

PRAGUE, CZECHIA
OCT. 5-7, 2026

Liveupdate IOMMU Feature Matrix and dependencies

Features

Single Stage IOMMU Persistence and Restore

DMA Allocation Preservation

Hitless Replacement of Domains to discard restored domains

ATS/Pasid Support

Page Request Interface

P2P Support (depends on VFIO dmabuf preservation)

viOMMU (Nested Translations)

Mutable IOMMU domains after preservation

Non VFIO kernel drivers

Platforms

Intel VT-d

AMD-Vi

Arm sMMUv3

Dependencies

IOMMUPT VT-d

Hitless Replacement (Intel, AMD and Arm)

IOMMUPT Arm sMMUv3

VFIO Selftests enhancements



LINUX

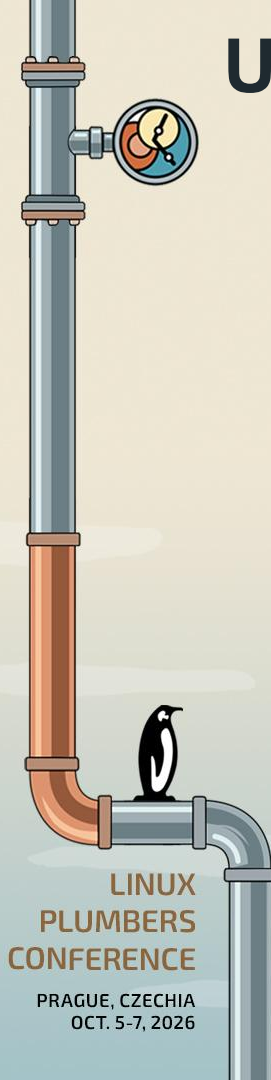
PLUMBERS CONFERENCE

PRAGUE, CZECHIA / OCT. 5-7, 2026



Upstream Status

- IOMMU Core + Intel Support
 - <https://lore.kernel.org/all/20260808022723.3893618-1-skhawaja@google.com/>
- Intel
 - Baolu working on hitless replacement of domains.
- Arm sMMUv3
 - IOMMUPT:
https://lore.kernel.org/all/0-v2-563ee63886f0+1209-iommupt_armv8_jgg@nvidia.com/
 - Liveupdate:
<https://lore.kernel.org/all/20260929071950.2710070-1-praan@google.com/#t>
 - Hitless Replacement:
<https://lore.kernel.org/all/20260925011308.3381953-1-skhawaja@google.com/>
- Dependencies
 - DMA Direct Allocation Preservation:
 - <https://lore.kernel.org/all/20260708234854.4044652-1-skhawaja@google.com/>
 - VFIO selftest enhancements
 - <https://lore.kernel.org/all/20260505221518.619123-1-skhawaja@google.com/>



LINUX
PLUMBERS
CONFERENCE

PRAGUE, CZECHIA
OCT. 5-7, 2026

Arm SMMUv3: A Primer

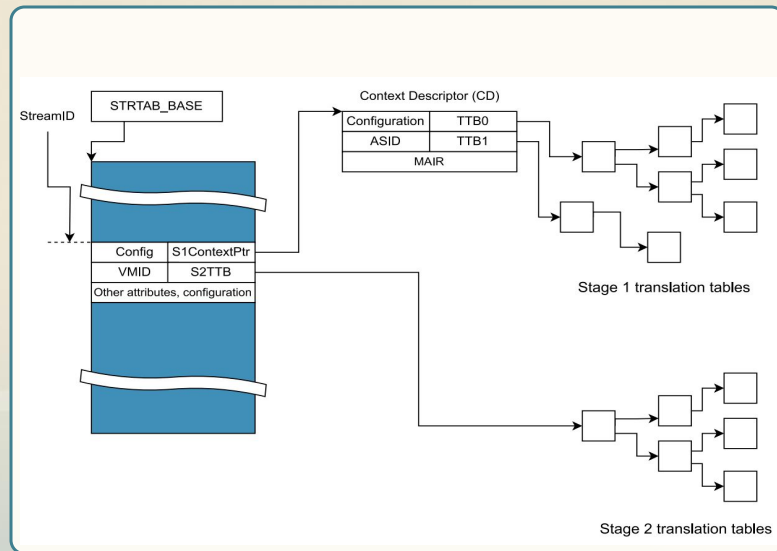
Tables: in memory, walked by HW

- Stream Table: STE per StreamID (device)
- CD: per address space, tagged by ASID
- Page tables: TLBs tagged by ASID / VMID

HW-Shared Queues: PROD/CONS in MMIO

- CMDQ (SW → HW): CFGI / TLBI / SYNC
- EVTQ (HW → SW): faults, config errors
- PRIQ (HW → SW): PCIe page requests

CR0.SMMUEN=1 ⇒ STRTAB_BASE is frozen



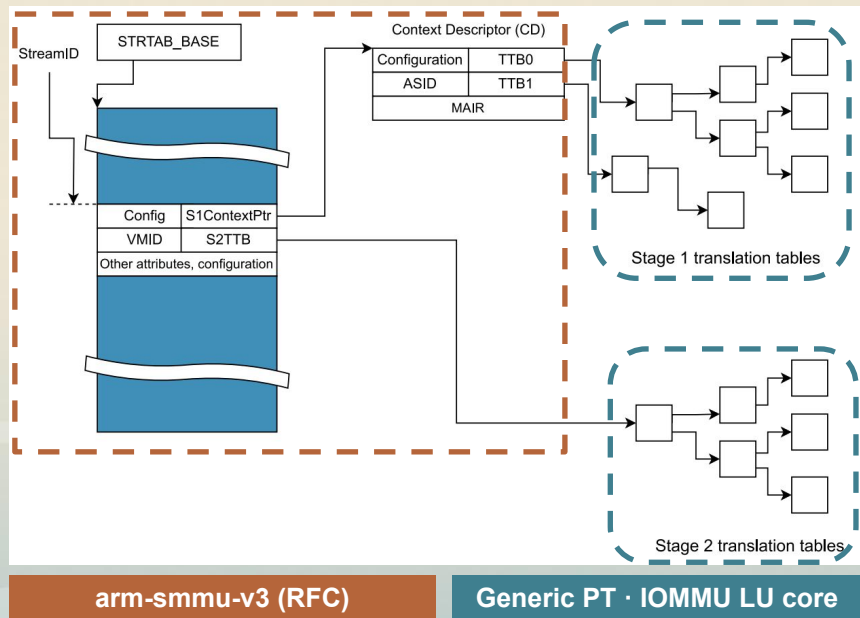
LINUX

PLUMBERS CONFERENCE

PRAGUE, CZECHIA / OCT. 5-7, 2026



What SMMUv3 needs to preserve



Preserved across the kexec

- CR0.SMMUEN stays 1
- Stream Table: L1 + L2s with preserved SIDs
- CD tables: S1, SSID 0
- ASID / VMID: recorded, then reserved
- Page tables: Generic PT (LU core)
- Not preserved: CMDQ (reset), PRIQ (disabled)

Queues: Which ones survive?

The VM's device keeps doing DMA through the kexec.

- Faults (if any) during shutdown, can we quiesce?
- Keep the queue?
 - The new kernel adopts it
 - Drains the backlog.



LINUX
PLUMBERS CONFERENCE

PRAGUE, CZECHIA / OCT. 5-7, 2026



Outgoing kernel: let go without stopping DMA

Preserve (on LUO preserve)

.preserve (per SMMU, first device)

1. Record MMIO base (token) & STRTAB_BASE_CFG
2. Preserve the EVTQ
3. Preserve Stream Table (L1 + l2_states[] or linear)

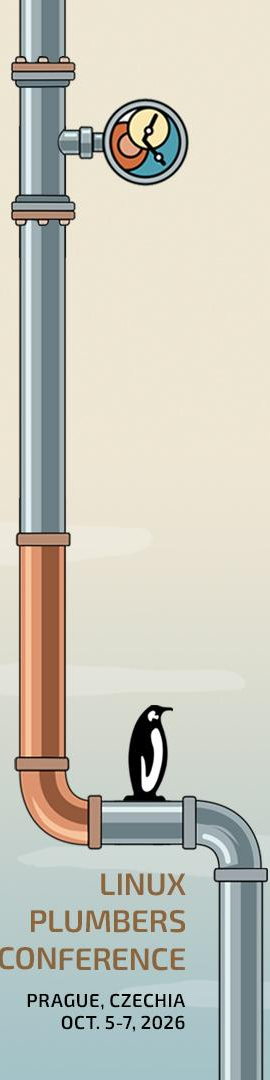
.preserve_device (per master)

1. Skip identity/blocked; reject SVA & active PASIDs
2. Record ASID (S1) or VMID (S2)
3. Preserve CD table (S1 only; nested in guest memory)
4. Preserve L2 Stream Tables holding master's SIDs
5. Mark master preserved

Shutdown (SMMUEN=1)

1. Mask IRQs; wait for a running EVTQ handler
2. Block unpreserved masters → abort STE
(maintains isolation for shared L2 tables that are preserved)
3. Unhook L2 tables with no preserved SID (clear the l1std)
(L2 memory isn't preserved & may be reused after kexec)
4. CFGI_ALL + TLBI_{EL2,NSNH}_ALL (syncd)
5. Disable CMDQ + PRIQ; persist SMMUEN + EVTQEN

On any failure, fall back to `arm_smmu_device_disable()`



Incoming kernel: take over, don't reprogram

1

Reuse, don't allocate

Stream Table, CD tables and EVTQ come back from KHO, validated against the live registers

2

Reserve IDs

Every ASID / VMID found in the live tables is reserved, so new domains never alias them

3

Hitless re-config

SMMUEN, STRTAB_BASE and the EVTQ are left alone; CMDQ / PRIQ reset as usual; latched GERROR acked

4

Catch up on events

IRQs are re-wired while masked, then the EVTQ thread is kicked to report events from the LU window

Discussion Topics

1. EVTQ: must it survive?

Faults abort regardless of logging. Preserving EVTQ only adds visibility into kexec-window faults and handover bugs.

Worth 1 MB for visibility, or disable and drop events?

2. IORT RMRs in kexec

LU shutdown keeps SMMUEN=1 and aborts unpreserved SIDs (including RMRs). Normal kexec clears SMMUEN and leaves DMA to GBPA.

Accept blackout, or keep RMR SIDs in bypass (preserving L2s)?

3. CMDQV: host VCMDQs

Incoming `device_reset` toggles `CMDQV_EN` without checking LU, resetting all VINTFs & guest queues.

Only host-owned state needs reset to clear stale state.

A: Reset, then restore guest VINTFs.

B: Skip reset: cycle VINTF0, adopt guest VINTFs.

Is VINTF0-only safe while guest VINTFs are up?

- **CMA:** Buffers > PAGE_SZ come from CMA; `dma-direct` refuses preservation. (EVTQ, L2 Stream Tables, CD leaves)
- Restore uses plain `iommu_paging_domain_alloc()`, so an S2 / NEST_PARENT domain comes back as S1
 - The core should preserve the flags and restore with them.

RFC: lore.kernel.org/all/20260929071950.2710070-1-praan@google.com



LINUX
PLUMBERS
CONFERENCE

PRAGUE, CZECHIA
OCT. 5-7, 2026

Getting off the restored domain: hitless replace

Re-attach is already hitless: the restored domain inherits the live ASID => CD/STE remain identical. But a restored domain is immutable (no map/unmap), so the VMM must replace it with a new HWPT.

arm_smmu_write_entry() is hitless only if exactly one qword of used bits changes

Entry	Qword A	Qword B	Result
S1 CD	q0: ASID $\neq$	q1: TTB0 $\neq$	2 qwords $\rightarrow$ V=0 $\rightarrow$ DMA faults
S2 STE	q2: VMID $\neq$	q3: S2TTB $\neq$	2 qwords $\rightarrow$ V=0 $\rightarrow$ DMA faults

1 · Temporary ID

Step through a temp ASID so each write changes one qword (Sami's RFC). S2: temp VMID.

2 · ID inheritance

The new HWPT takes over the restored ASID/VMID: only TTB changes $\rightarrow$ 1 qword + TLBI.

3 · In place

Make the restored domain mutable again, on top of the preserved page table.

Existing HWPT replace mechanism in iommufd can be used