IOASID Extensions for Scalable IO Virtualization (SIOV)

LPC 2020 VFIO/PCI/IOMMU MC

jacob.jun.pan@intel.com
yi.l.liu@intel.com

Purpose & Background

Ratify the design of proposed IOASID API extensions.

IOASID extensions for guest SVA

https://lore.kernel.org/lkml/1598070918-21321-1-git-send-email-jacob.jun.pan@linux.intel.com/

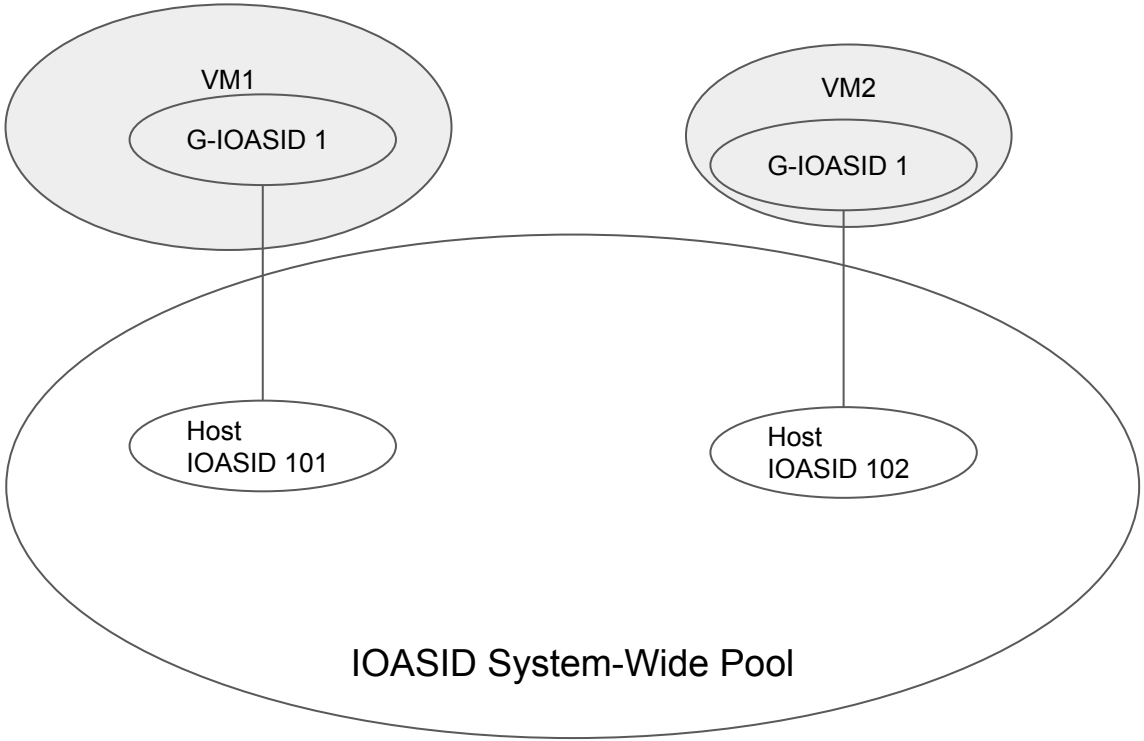
Background: IOASID is a generic kernel library (since v5.5) for managing PCI PASID and ARM SMMU SubstreamID

Problem Statement: Gaps in Existing APIs for vSVA Support

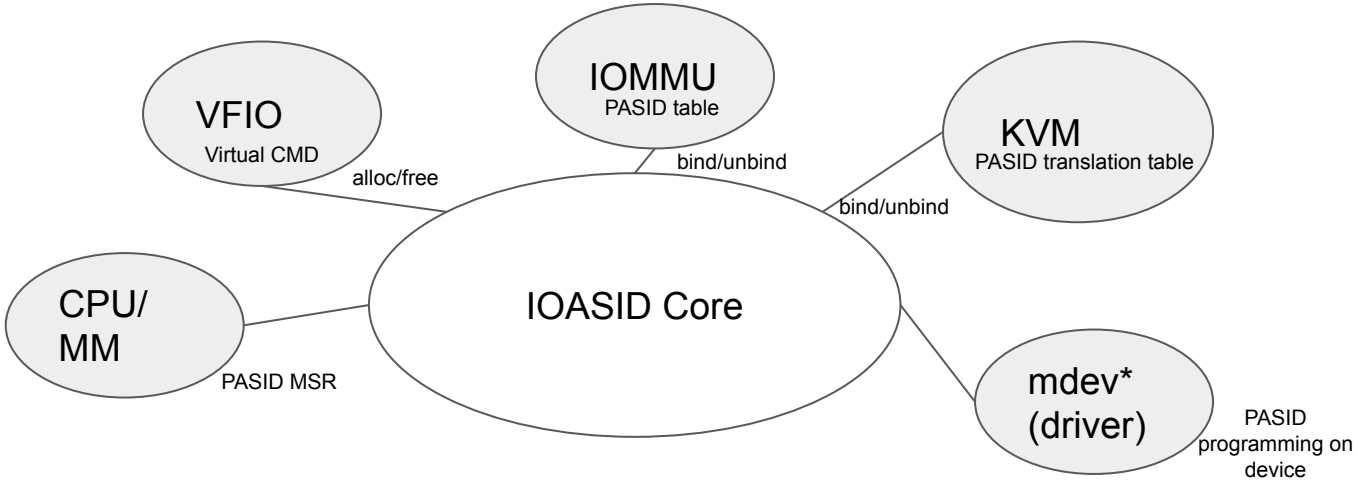
1. Provides basic allocation/free service based on XArray
2. Allocators can be customized, support VT-d virtual command interface for guest usage
3. Has IOASID_SET concept for basic group management
4. Manage IOASIDs by groups, e.g. enforce ownership, quota, etc.
5. Synchronize states among IOASID users, e.g. when IOASID is freed, unbound.
6. Support non-identity guest-host IOASID mapping
7. Manage lifecycle across many users

- Existing
- Gaps

Problem Statement: IOASID set



Problem Statement: Keep IOASID users in sync
Example: SIOV* platforms



*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

Proposed Solutions*

- Redefine and extend IOASID set such that IOASIDs can be managed by groups.

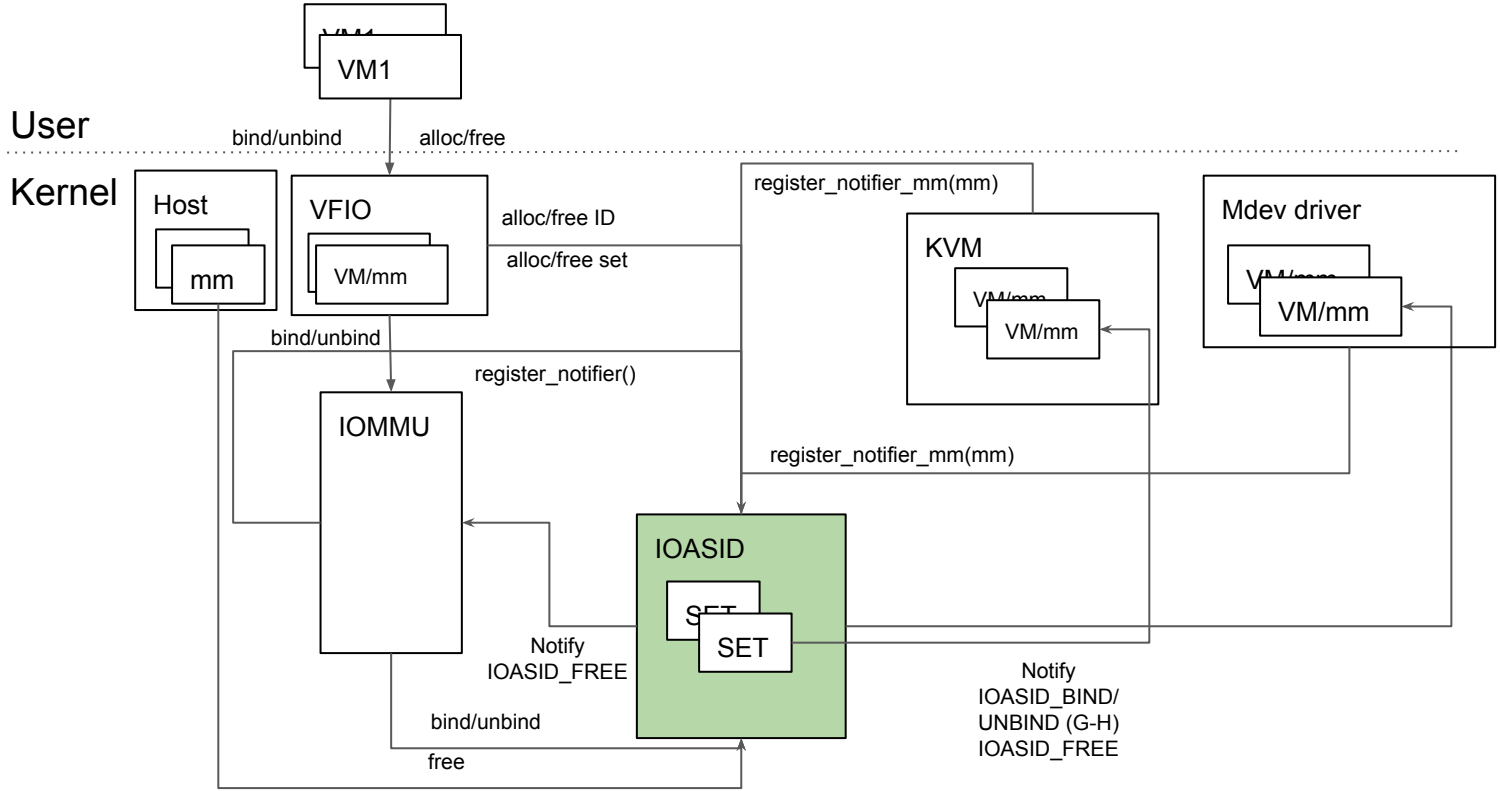
- Add notifications for IOASID state synchronization

- Add reference counting for life cycle alignment among users

- Add ioasid_set private IDs, which can be used as guest IOASIDs

*https://lore.kernel.org/lkml/1598070918-21321-1-git-send-email-jacob.jun.pan@linux.intel.com/

Intel Scalable IO Virtualization Usages



IOASID Set Key APIs

```
+ struct ioasid_set *ioasid_alloc_set(void *token, ioasid_t quota, u32 type)

+ int ioasid_adjust_set(struct ioasid_set *set, int quota);

+ void ioasid_set_get(struct ioasid_set *set)/void ioasid_set_put(struct ioasid_set *set)

+ int ioasid_set_for_each_ioasid(struct ioasid_set *set, void (*fn)(ioasid_t id, void *data), void *data)
```

IOASID API

```
+ ioasid_t ioasid_alloc(struct ioasid_set *set, ioasid_t min, ioasid_t max, void *private);

+ int ioasid_get/put(struct ioasid_set *set, ioasid_t ioasid);

+ void *ioasid_find(struct ioasid_set *set, ioasid_t ioasid, bool (*getter)(void *));

+ ioasid_t ioasid_find_by_spid(struct ioasid_set *set, ioasid_t spid)

+ int ioasid_attach_data(struct ioasid_set *set, ioasid_t ioasid, void *data);

+ int ioasid_attach_spid(struct ioasid_set *set, ioasid_t ioasid, ioasid_t spid);
```

IOASID Notifier API

```
+ int ioasid_un/register_notifier(struct ioasid_set *set, struct notifier_block *nb)

+ int ioasid_un/register_notifier_mm(struct mm_struct *mm, struct notifier_block *nb)

+ int ioasid_notify(ioasid_t ioasid, enum ioasid_notify_val cmd, unsigned int flags)
```

Questions?

BACKUP: IOASID Lifecycle in Guest SVA

```
  VFIO        IOMMU        KVM        VDCM        IOASID       Ref
.....................................................................
1             ioasid_register_notifier/_mm()
2 ioasid_alloc()                                                  1
3 bind_gpasid()
4             iommu_bind()->ioasid_get()                          2
5             ioasid_notify(BIND)
6                          -> ioasid_get()                        3
7                          -> vmcs_update_atomic()
8 mdev_write(gpasid)
9                                  hpasid=
10                                 find_by_spid(gpasid)           4
11                                 vdev_write(hpasid)
12 -------- GUEST STARTS DMA --------------------------
13 -------- GUEST STOPS DMA ---------------------------
14 mdev_clear(gpasid)
15                                 vdev_clear(hpasid)
16                                 ioasid_put()                    3
17 unbind_gpasid()
18            iommu_ubind()
19            ioasid_notify(UNBIND)
20                         -> vmcs_update_atomic()
21                         -> ioasid_put()                        2
22            ioasid_put()                                        1
23 ioasid_free()                                                  0
24                                                  Reclaimed
-------------- New Life Cycle Begin -----------------------------
1  ioasid_alloc()                                    ->           1
```