

# Libmetal Shared Memory Allocation

Wendy Liang



# Issue to Solve

- > **We want to build libmetal abstraction to enable application to allocate memory to shared data with a device or remote processor.**
- > **With libmetal abstraction, the application doesn't need to care about the underline memory management across different operating systems, platforms and memories.**

# Use Cases

## > Here are the use cases we are considering for now:

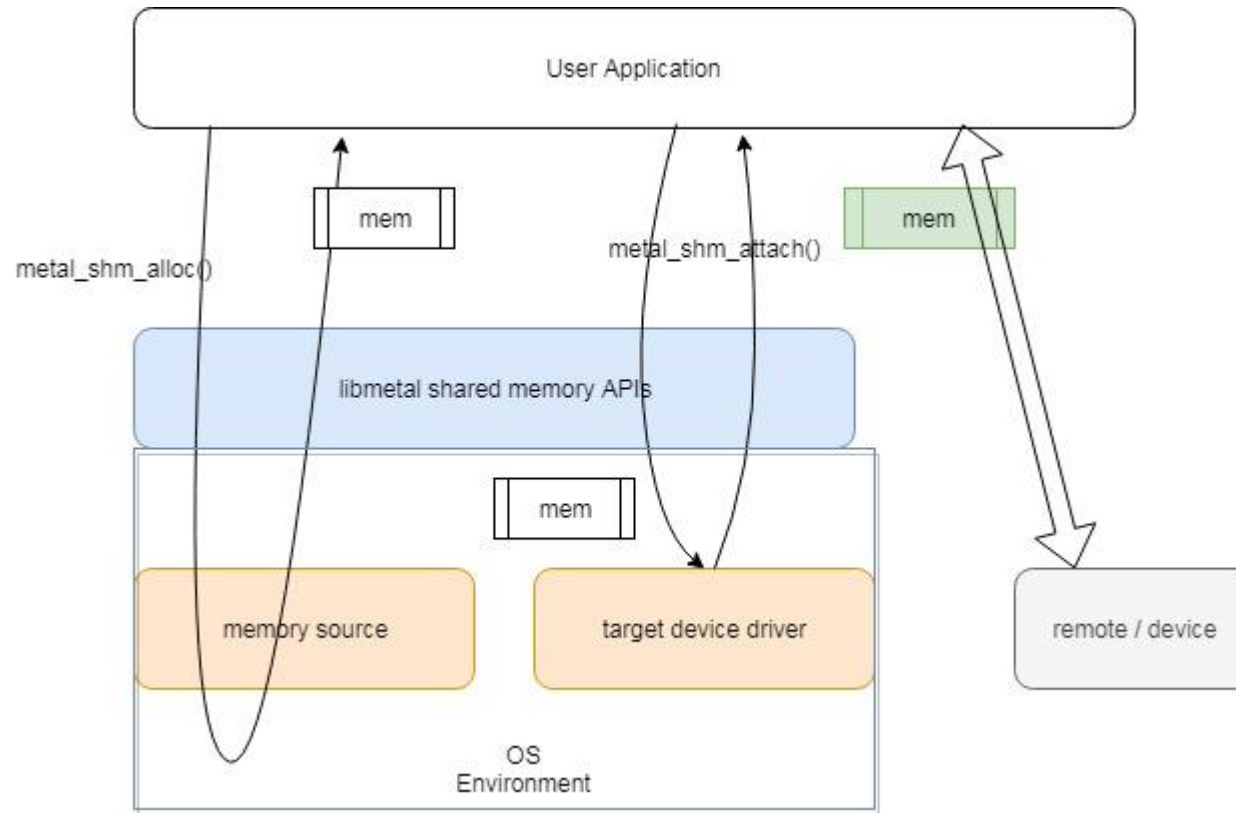
### >> UC1:

- Linux/Baremetal application without IOMMU needs to share DDR memory to a device/coprocessor.

### >> UC2:

- Linux application with IOMMU enabled needs to share DDR memory to a device/coprocessor.

# New Shared Memory APIs



# New Shared Memory APIs - Description

## > Shared memory allocation API

- >> Function to allocate memory from a device (e.g. memory device)
  - `struct metal_generic_shm *metal_shm_allocate(size_t size, struct metal_device *src_dev, metal_shm_type shm_type);`

## > Shared memory attachment API

- >> attach the shared memory to the destination device
- >> this operation will make the shared memory available to the remote. (It may map the memory differently to the source device in order to have the destination device to access it.)
- >> The API will also return the memory device address which application can be used to setup DMA or notify remote.
  - `struct metal_generic_shm *metal_shm_attach(struct metal_generic_shm *src_shm, struct metal_device *dst_dev);`

## > Shared memory detach API

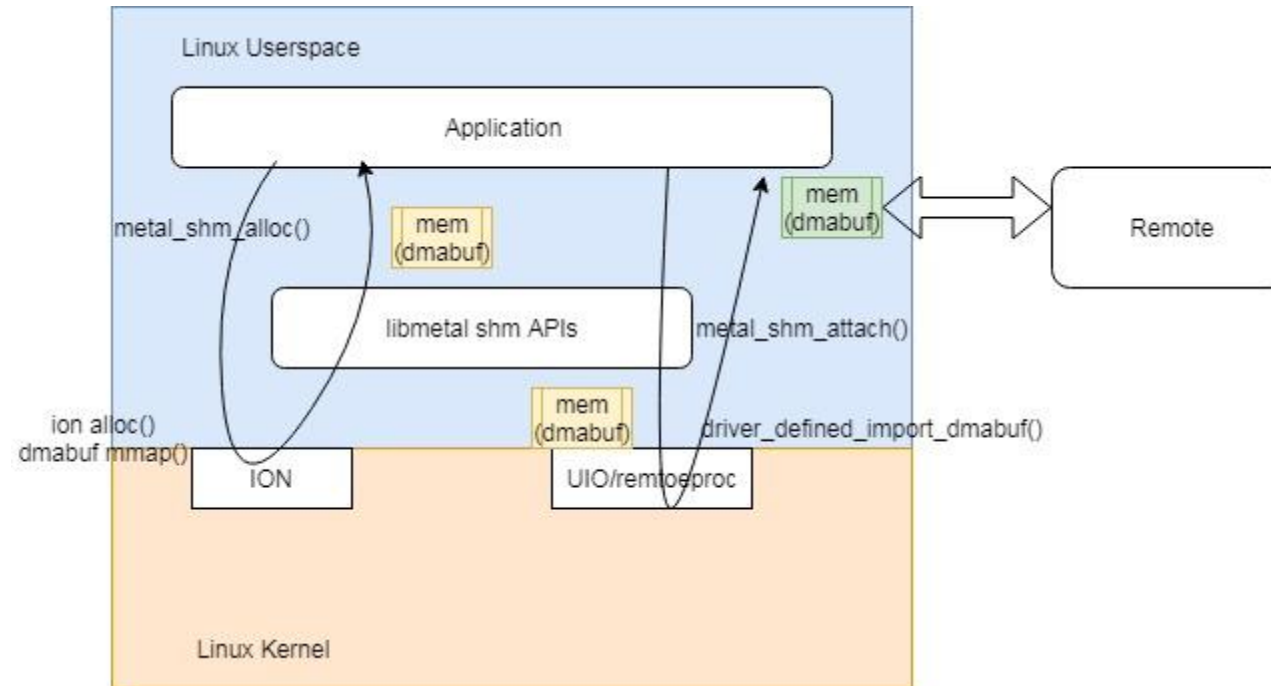
- >> detach the memory from the destination device
  - `void metal_shm_detach(struct metal_generic_shm *dst_shm, struct metal_device *dst_dev);`

## > Shared memory Free API

- >> free the memory from the source device
  - `void metal_shm_free(struct metal_generic_shm *src_shm, struct metal_device *src_dev);`

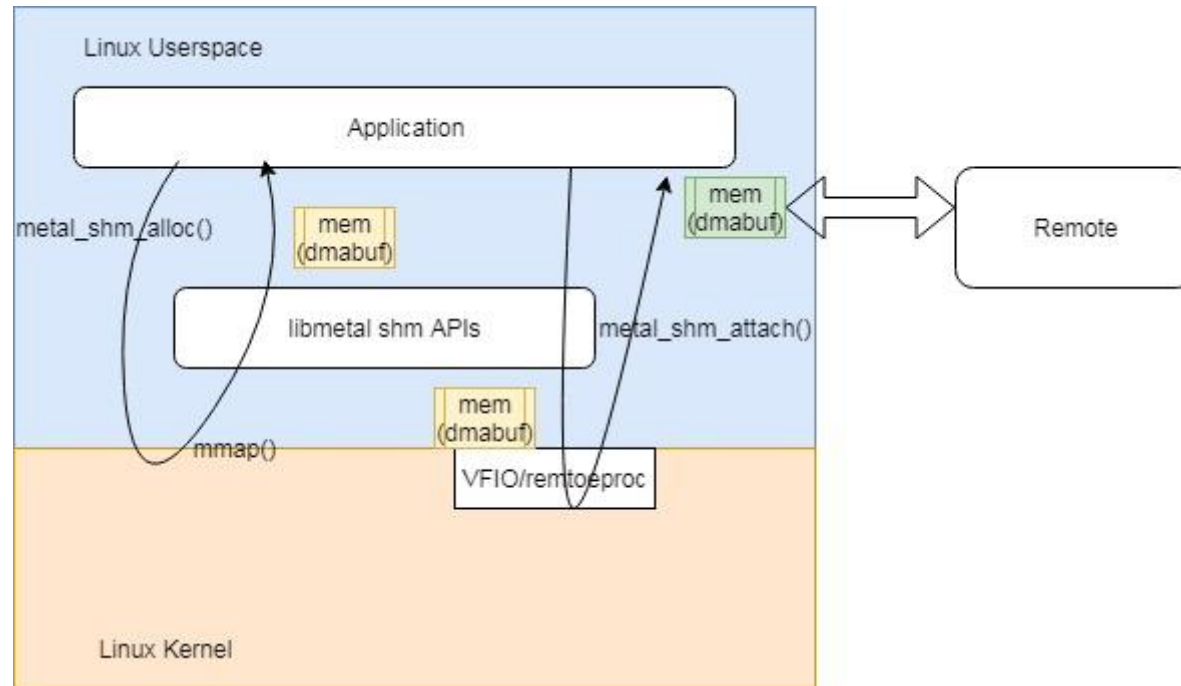
# Libmetal Shared Memory Allocation For Linux

## > Without IOMMU:



# Libmetal Shared Memory Allocation For Linux

## > With IOMMU:



# Libmetal Shared Memory Allocation For Linux

## > Without IOMMU

- >> Allocate shm with `metal_shm_allocate()` from source device.
  - e.g. the source device can be ION device. It gets the DMA buf from the device.
  - We consider to use ION device so that libmetal can have generic implementation to get DMA buffer
- >> Attach shm to the target device with `metal_shm_attach()`
  - the target device Linux kernel driver will import the DMA buf and map it to make sure it is available for the remote.
  - We propose to add importing DMA buffer to UIO driver, so that the libmetal for Linux can have generic implementation for DMA import.

## > With IOMMU

- >> Allocate shm with `metal_shm_allocate()`.
  - This step will allocate some space with `mmap()`.
- >> Attach shm with the target vfio device with `metal_shm_attach()`.
  - This step will call `vfio iotl` to map the memory for the device.



# Github Libmetal Issue to Track

> <https://github.com/OpenAMP/libmetal/issues/70>