

Partial cache flush use-cases for DMA-BUFs

Codec2 use-case in Android 11

- Before Android 11, buffer allocation was done by codecs.
- For streaming applications apps,
 - downloaded the content,
 - parsed it, and
 - copied the data to be decoded into buffers allocated by codec and queued them to be decoded.

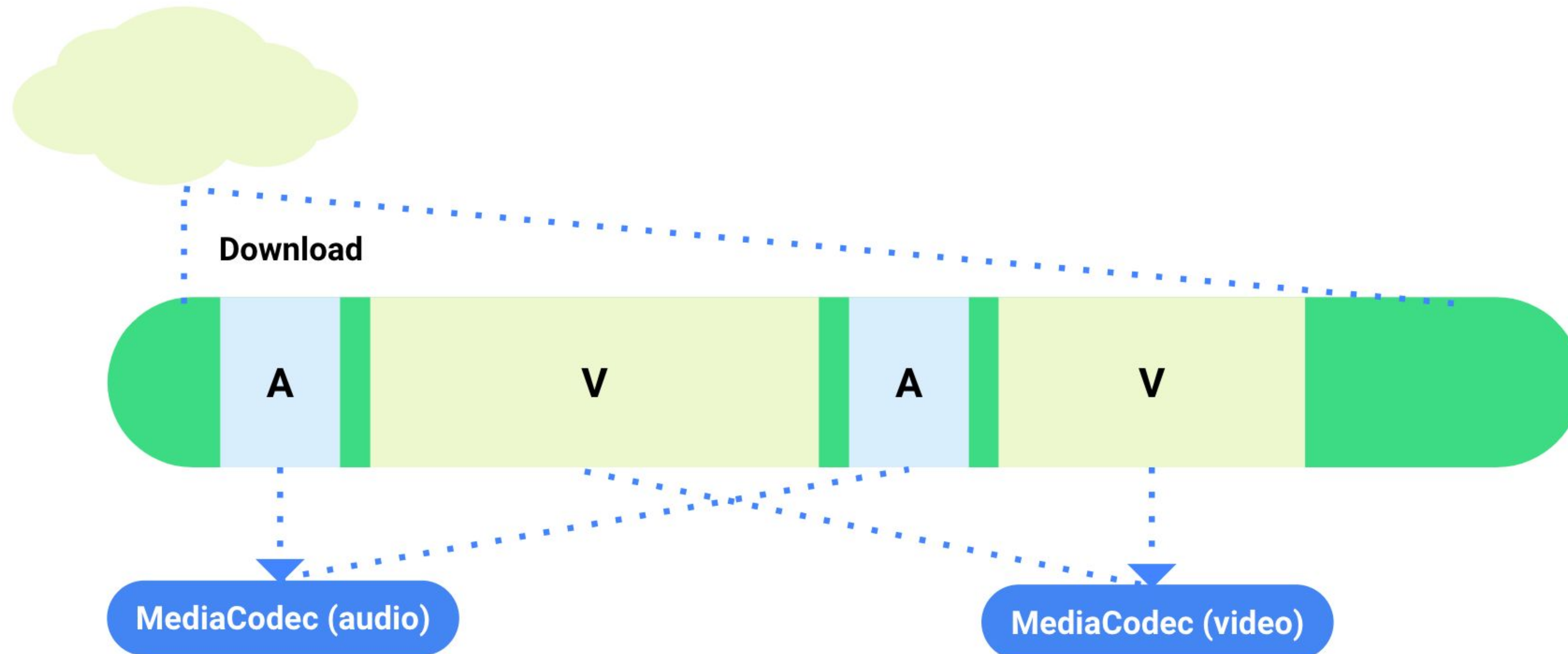
Codec2 use-case in Android 11

- Before Android 11, buffer allocation was done by codecs.
- For streaming applications apps,
 - downloaded the content,
 - parsed it, and
 - copied the data to be decoded into buffers allocated by codec and queued them to be decoded.

- Android R allows codecs to be **configured to accept buffers allocated by apps**.
 - Apps allocate the buffer(or system may recycle one of the previously allocated buffers),
 - map the buffer and download content into it,
 - parse it and queue the desired range of the buffer to the codec.
 - The component receives the handle to the entire buffer and the offset and size that denotes the range.
 - The component must now do a **cache flush for the range** before proceeding to read the range and processing it.

Codec2 use-case in Android 11

Example: Decoding Session



Codec2 use-case in Android 11

- Eliminates the need for a separate buffer to hold the downloaded content.
- Savings add up for 4K/8K content.

Other use-cases

- Metadata flushing.
 - Frame Buffer Compression metadata headers which indicates compression format for block.
 - Memset insufficient to reset the metadata.
 - Need to write initialization values into the header and cache flush.
 - Potential performance improvement with partial cache flush.

Solution to allow parallel use of different regions on the same buffer

- Add `begin_cpu_access_partial()` and `end_cpu_access_partial()` hooks to `dma_buf_ops`.
- Similar to `begin/end_cpu_access()` but with additional offset and length parameters to specify range
- AOSP implementation [here](#)
- Add new `DMA_BUF_IOCTL_SYNC_RANGE` ioctl that takes start/end ranges



Similarly for the proposed device usage annotations..

- Add `begin_device_access_partial` and `end_device_access` partial hooks to `dma_buf_ops`.
- Solution similar to range locks on files.



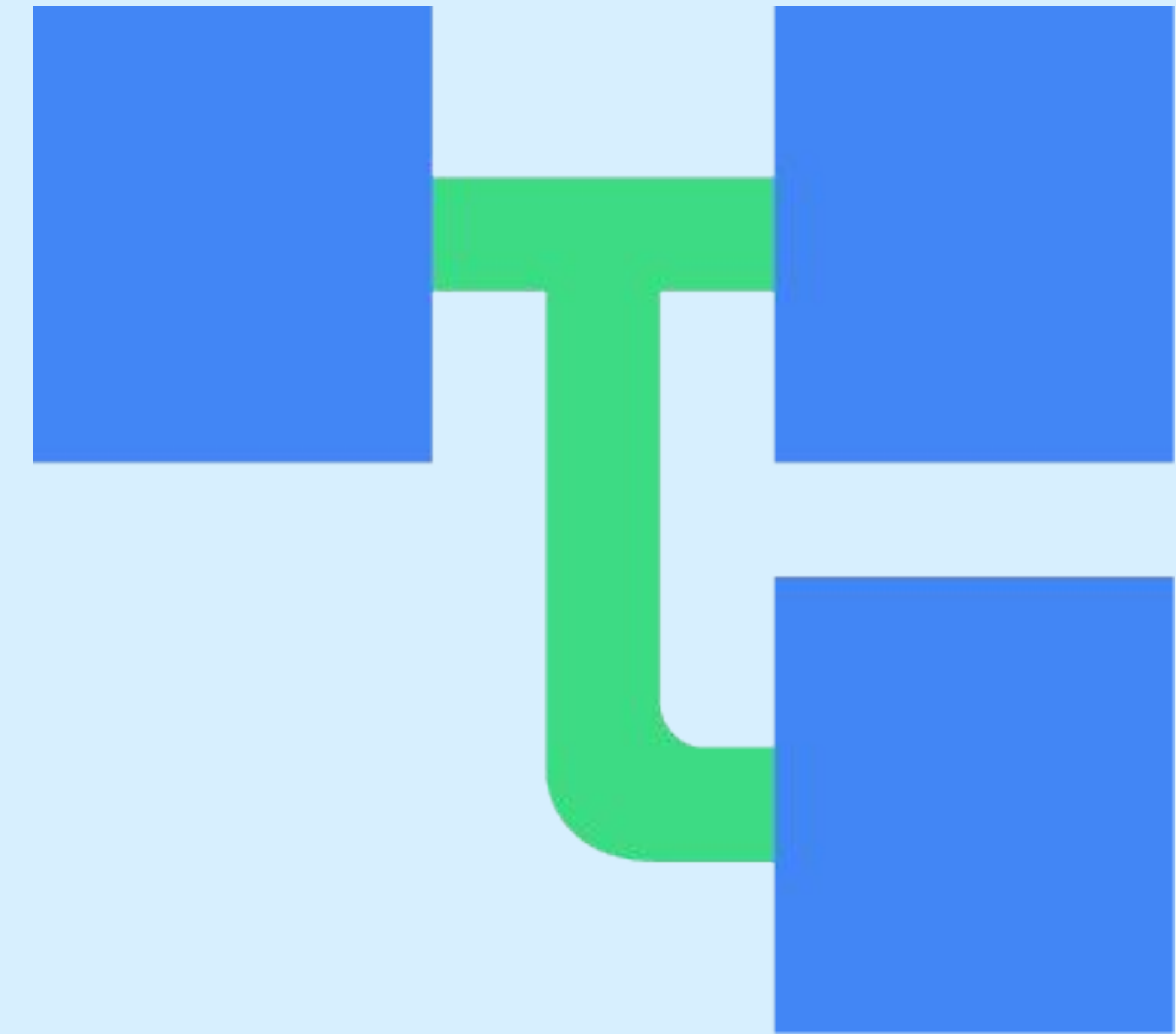
Challenges

- Complicates the question of buffer ownership.
- Ownership needs to be tracked range-wise.
- Complicates the proposed lazy-flushing optimization.



Discussion topics:

- Any other use-cases?
- Alternate solutions?
- Deal-breakers?



Thank you!

Backup Slides

Related optimization

- IOCTL to only cache sync the memory mapped by the CPU (codeaurora implementation by Liam Mark [here](#)).