V4L2 M2M EXT API enhancement

Jun 2023 Hsia-Jun(Randy) Li

synaptics

Quick

1. How to answer those allocation requirement for planes in a memory?
2. Do we need a new API for format enumeration?
3. vb2_mem is not enough for a complex memory allocation
4. Secure(Digital Copyright) Video Pipeline
5. Many drivers are occupying large memory without user acknowledge!
6. V4L2 schedule and notification

How to answer those allocation requirement for planes in a memory?

synaptics

Where goes M-variant pixel format

Although all mainline DRM devices support address for each buffer plane

- Low cost device may not have IOMMU and request contiguous memory
- We still didn't sync the alignment requirement between V4L2 and DRM
- Those powerful GPU drivers would like NOT allocate memory from kernel dumb API but vendor API and GBM

We need to make DRM export the alignment and cache hints then use them in CREATE_BUFS

Should we merge the REQBUF and CREATE_BUF

- struct v4l2_requestbuffers can't set the v4l2_format
- struct v4l2_create_buffers lacks of property for cache hints
- REQBUF didn't know the maximum buffer that a queue need
- The case REQBUF could do is recall all the buffers(count = 0)

Do we need a new API for format enumeration

synaptics

Pixel formats and colorspace enumuation

VIDIOC_ENUM_FMT would return only one fmtdesc each time

- DRM would return a fmt blob id for userspace mapping and parsing
- A pixel with a base format and a tiled variants - enumeration costs time

Colorspace would not change if no pixel format transform

- Decoding won't care about colorspace
- Four members of a colormetry have too many combines

Many drivers are occupying large memory without user acknowledge!

synaptics

The buffer hiding behind drivers

The reconstruction buffer in encoder

The composition or scaling buffer in the decoder

- VIDIOC_S_SELECTION
- VIDIOC_S_FMT - The client may choose a different format than selected/suggested by the decoder in VIDIOC_G_FMT()
- VIDIOC_S_FMT - it may be different if the hardware supports composition and/or scaling

AV1 film gain is a after decoding processing

Why we need the reconstruction bufffer

- reconstruction is not the same buffer of input buffer
  - human visible quality lower
- reconstruction could be a different image pixel format than the input which is optimized more for the hardware
- Scalable Video Coding
  - each layer is independt, stateless encoder would have the free to choose which buffer(reconstruction) as its referring frame(inter).

Why decoder need extra buffers for scaling

A frame(decoded result) could be used for referring later(not B frame)

Cropping could lead pixel layout difference especially in tiled pixel format

- linear pixel buffer may go pixel stride for right direction cropping

The pixel format that applicant requested may be inefficient

- Hardware could use its internal buffer for inter composition

DRM doesn’t have a queue(we don’t count ping-pong)

DRM write-back won’t redraw the “out” buffer many times

- It is a way to implement async page flip - GPU would redraw (partial) of input graphics

Secure(Digital Copyright) Video Pipeline

synaptics

Secure video pipeline

Regisers can’t be access directly

Buffer address is a useless information to the user

We need the blow features to make it work in V4L2

Demux or bitstream format transform

- Its capture buffer may reuse the same buffer as output
- It would be better we make those _SLICE and NO_SC as a modifier

decoding

- REE(linux kernel) has less control in its capture buffers allocation *(detail later)

metadata

- Need the DMA output buffer - subtitle or color enhanced info could be large

vb2_mem is not enough for a complex memory allocation

synaptics

V4L2 mem_ops allocator private data

MMAP usually means allocate from main memory for the most edge platforms

- This means DMA-Heap for the most platforms
- OUTPUT queue and CAPTURE queue could use a different DMA-heap
- Each plane is a buffer may use a different DMA-heap

V4L2 memory API change requests for secure codec

vb2_mem_ops->alloc()

- Need to know the cache hints and alignment - M variant is gone
- Need to know the pixel formats - Different pixel format could have a different cache setting
- Need to know the (hardware) pixel width and height - they are a part of cache hints

vb2_mem_ops->reclaim() * New API or except steps for vb2_mem_ops->prepare()

- Different pixel format or resolution could use a different pixel hints
- We may merge two or more buffers for one buffer

Allocator private

Buffers in a queue could have a different sizes or pixel formats

Import a DRM frame id

There are many non standard secure information how to pass by application, ex: secure session id

In Android tunnel video mode, userspace can't control pixel formats or source resolution

- What the userspace or we said REE world could control is which buffer would be presented(by PTS) - That is under the Android vendor media layer

DMA-FD could only pass an address, which is not enough for a secure memory

V4L2 schedule

synaptics

V4L2 m2m running without buffer

We still have old drivers that don't use m2m_ops->device_run

- Dynamic Resolution Change - No CAPTURE buffer in setup should be regarded as a transition or it break our V4L2 job scheduler

We need a hardware occupied queue beyond ready_queue

- Many drivers have its private buf (vb2_queue->buf_struct_size) to track this state
- It also wastes of time to loop the ready_queue in each woker
- VPx Codecs have non-display frame - it can't be in ready_queue as the future decoding may use it
- V4L2 need to schedule hardware for such worker

V4L2 notification

synaptics

V4L2 M2M event

Notification the user a non-display frame is decoded

- Temporal unit - bad idea
  - introducing latency - who know when its show_frame flag comes, the HW is in idle!!!
  - Which non-display frame can't be decoded when errors in bitstream
- They are never in the display order

resolution changes

- The last buffer from before the change must be marked with the V4L2_BUF_FLAG_LAST flag, similarly to the Drain sequence above
  - We the resolution changes happens we may don't have any queued buffers in the CAPTURE;
  - Then we could to send out a buffer with bytesused = 0 with invalid index.
  - If we still have queued CAPTURE buffer, are we removing it from the m2m ready queue?
- Super frame or golden frame requests a larger buffer

EOS

- User may need those frames which theirs display order didn't come

V4L2 M2M event

Subscribe event is a bad idea for those events that occurring time is matter

- Android OpenMAX developers have a lots of pains here

Event should go with buffer - before or after a buffer

- We could send out a buffer with an invalid index to indicate it is for an event

Should We have the third queue?

reconstruction buffer

scaling or composition buffer

extra buffers for a codec context

- User should have the right to choose which memory should be used - Android way

Thank You

Hsia-Jun(Randy Li)

synaptics