dri2video

Rob Clark

Texas Instruments

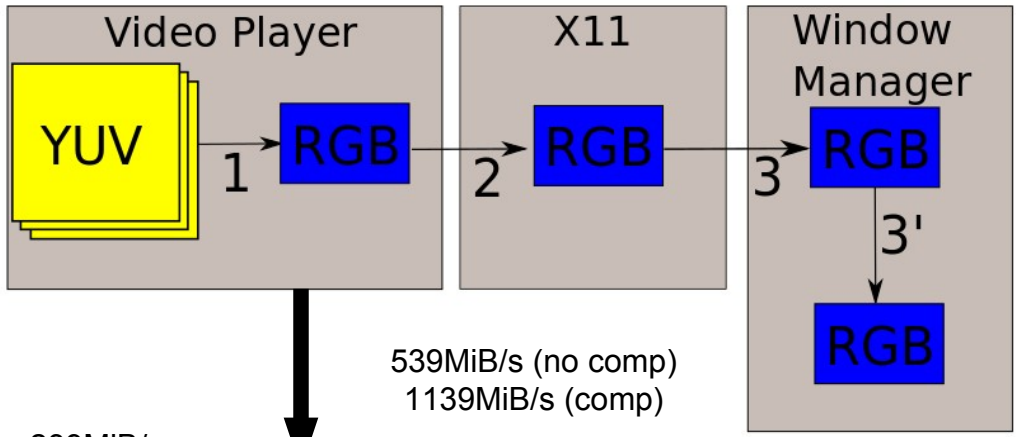
X11 – Video Rendering

- Traditionally Xv extension used for rendering video
  - Xshm buffers: 2x memcpy
    - Not terribly good for hw decoders that have special memory requirements
    - And not terribly good for GPUs either.. need a copy into a GPU accessible buffer or at least map/unmap on every frame (userptr)
- DRI2
  - Used under the hood by VAAPI/VDPAU.. but can only support unscaled RGB buffers, so GPU blit YUV->RGB + scaling done on client side

X11 – dri2video

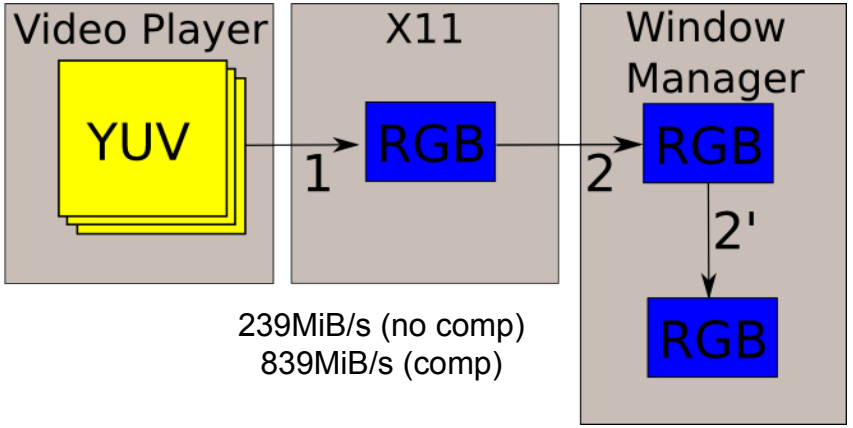
Example memory bandwidth savings based on 1080p 30fps NV12 video rendered to nearly fullscreen window on 1280x1024 display

NV12->RGB = (1920*1080*1.5) + (1280*1024*4) → 239MiB/s
Swap/blit = (1280*1024*4) * 2 → 300MiB/s
Composite = (1280*1024*4) * 2 → 300MiB/s
Presentation blit = (1280*1024*4) * 2 → 300MiB/s



539MiB/s (no comp)
1139MiB/s (comp)

NV12->RGB = (1920*1080*1.5) + (1280*1024*4) → 239MiB/s
Swap/blit = (1280*1024*4) * 2 → 300MiB/s
Composite = (1280*1024*4) * 2 → 300MiB/s
Presentation blit = (1280*1024*4) * 2 → 300MiB/s



239MiB/s (no comp)
839MiB/s (comp)

X11 – dri2video

- Combines the ideas of Xv and DRI2
- Xserver (DDX driver) allocates GEM buffer and passes to client process
  - Allows to use VRAM or deal w/ any other special memory requirements
- But unlike DRI2, the buffer can be YUV (incl. Multi-planar), sized according to video size, not scaled drawable size, and cropped
- Can support zero-copy overlays too if display can scanout GEM buffers
- Should be helpful for other hw decoders: VAAPI, etc
  - Esp. UMA setups where the extra blit is consuming system memory bandwidth

Proto – New messages

- DRI2GetBuffersVid
  - Like DRI2GetBuffersWithFormat but adds width/height
- DRI2SwapBuffersVid
  - Like DRI2SwapBuffers but adds src crop coords
- DRI2SetAttribute / DRI2GetAttribute
  - Analogous to Xv Get/SetPortAttribute
    - CSC matrix, etc
- DRI2GetFormats
  - Get supported fourcc formats
    - Recommendation: use at least one '\0' or non 7-bit ascii character for custom formats
- DRI2BUFFER
  - Add extra name/pitch (for multi-planar formats)

Things missing

- Interlaced / Stereo
- Client process control over single buffer vs buffer per plane
- Destination coordinates
  - Preserve aspect ratio independent of dest drawable dimensions?
- Possible Idea
  - Make traditional dri2 and dri2video attachment points non-overlapping
    - Fallback to dri2 for anything dri2video can't do?