BoF: Negotiating DMA-BUF Heaps (and other discussions)

Ezequiel Garcia
Taking some notes

• https://pad.riseup.net/p/bof-dma-buf-heap-lpc-2020
Current DMA-BUFs exporters

- Subsystem specific
  - Video4Linux2 API (aka Videobuf2)
  - GEM API
- DMA-BUF Heaps
- No mechanism to actually expose device constraints and negotiate mappings parameters.
Questions

- In-kernel DMA-BUF Heap interface
  - Should this replace subsystem-specific interfaces?
  - Should new subsystems avoid messing with allocating DMA-BUFs?

- Device constraints and heap capabilities
  - aka DMA-BUF “negotiation”
In-kernel heap interface

- Is this a good idea?
- Should this replace subsystem-specific implementations?
- Should new subsystems avoid messing with allocating DMA-BUFs?
Negotiating heaps

Daniel Vetter

> the rough idea is that in sysfs every device lists all the heaps it can use, and then you pick the common one that works for all devices.

https://www.spinics.net/lists/dri-devel/msg267882.html
Negotiating heaps

Laurent Pinchart

> Devices (would be) exposing constraints, and allocators
> exposing parameters (capabilities?), with a userspace
> library to reconcile the constraints and produce
> allocator parameters from them.

https://www.spinics.net/lists/linux-media/msg175613.html
Negotiating heaps

• Constraints (draft)
  – pitch
  – offset alignment
  – cache coherency
  – physical memory bank placement
  – iommu presence
  – Should these be opaque?
  – How are these exposed?
Thank you!