ION/DMA-BUF Heaps Transition & DMA-BUF Cache Handling

John Stultz <john.stultz@linaro.org>

Linaro

DMA-BUF Heaps

- Landed in 5.6 (with system and CMA heaps)
  - Credit: **Rebecca Schultz Zavin** for creating ION and **Laura Abbott** for maintaining ION
  - + Tons of other contributors over the years
- Starting to get some vendor interest in transition from ION
- Want to pull vendors into the community on this,
  - Want common shared heaps, rather than a collection of mostly out-of-tree vendor-specific heap implementations as was common with ION
- Performance is going to be key!

- Skeleton Heap Driver:

  https://git.linaro.org/people/john.stultz/android-dev.git/log/?h=dev/dma-buf-heap-skeleton

- Android transition helper library:

  https://android.googlesource.com/platform/system/memory/libdmabufheap/

Work in Progress/Topics for Discussion

- AOSP Codec2 userland transition
- Uncached heap(s) (Missing ION_FLAG_CACHED)
- Exposing multiple CMA areas (Kunihiko Hayashi - needs in-tree user!)
- Heaps as modules support (needs in-tree users!)
- In-kernel allocation accessors (needs in-tree users!)
- Optimizations! (but which ones really matter to vendors?)

- Most importantly: PARTICIPATION!

Work in Progress/Topics for Discussion

- AOSP Codec2 userland transition
- Uncached heap(s) (Missing ION_FLAG_CACHED)
- Exposing multiple CMA areas (Kunihiko Hayashi - needs in-tree user!)
- Heaps as modules support (needs in-tree users!)
- In-kernel allocation accessors (needs in-tree users!)
- Optimizations! (but which ones really matter to vendors?)

- Most importantly: PARTICIPATION!

Just in the last week!

- Device DMA Heap (Ezequiel Garcia - per-device heaps?)
- Chunk Heap (Hyesoo Yu - CMA based chunk heap)

DMA-BUF Cache Handling LWN articles

- LWN Article:
  - Part 1: https://lwn.net/Articles/822052/
  - Part 2: https://lwn.net/Articles/822521/

- Summary
  - Parallel-mappings breaks DMA “ownership” transfers on map/unmap
  - Extraneous cache management operations needlessly cost performance
  - Need to move to consistent & explicit semantics for “ownership” tracking for DMA-BUF exporters in order to allow for optimal cache-maintenance/performance

What to do?

My proposal

- Adding `begin_device_access()` and `end_device_access()` calls and hooks in the `dma_buf_ops.`
    - Allows us to properly annotate/track ownership & cache domain transfers
    - Can do CPU cache operations lazily on transitions only
- Eventually, move ownership tracking and CPU cache handling to dmabuf core?
- Add lockdep style correctness checks?

Complexities:

- Device to Device hardware fence chains (no driver interactions in between)
    - Buffer goes in to the chain and comes out, so book end at those points (begin_device_access/end_device_access may not have to pair on same device)
- Complicates partial buffer cache flushing
- Graphics folks don’t seem to like rules/constraints :)

Thanks!

More discussion @ the Android MC BoF
Weds 7pm PDT