Panthor

Status update

COLLABORA

arm

What is Panthor?

- Upstream kernel driver for Arm Mali GPUs that are using the new CSF architecture (Arch10 and onwards)
- Based on Panfrost, that supports Arm Mali GPUs prior to Arch10
- Developed by Collabora with involvement from Arm GPU software group

Who are we?

- Boris Brezillon
  - Original author of the driver, working for Collabora, maintainer
- Daniel Stone
  - Mesa developer, CI, working for Collabora
- Liviu Dudau
  - Initiator of an upstream driver for Mali CSF GPUs, switched to Panthor once the first revision was published, working for Arm, maintainer
- Florent Tomasin
  - GPU kernel driver tech lead at Arm

arm

2025: Consolidating our roots

- A bunch of interesting and relatively self-contained features
  - Added support for realtime priority for groups (Mary)
  - Added some initial memory statistics using fdinfo (Adrián)
  - BO labelling and tracking of GEM objects through the DebugFS interface (Adrián)
- The usual set of fixes
  - Improvements in the robustness for suspend/resume (Boris)
  - MMU, cache coherency fixes (Boris, Akash)
  - A gazillion of other fixes
- Preparing for new hardware
  - Introduced a small internal abstraction layer to prepare for future architectural changes in the GPUs (Karunika)

2026: Becoming a grown-up driver

- Sparse GEM population (Adrian, Boris)
  - Or how to fix a long standing issue in Mali drivers (Lima, Panfrost and Panthor)
  - Discussion started on the ML
  - Led to the writing of a piece of doc around tilers and why on-demand allocation is a problem
  - Blocked on us (still have to post of new version of each of those patch series)
- VM_BIND/GPUVM repeated mappings (Caterina, Boris)
  - Needed to efficiently support sparse bindings
  - Initial version posted on the ML, based on some prior patches from Lina (Asahi)
- Memory shrinker (Liviu, Chris)
  - Aiming for a transparent reclaim model like Xe
  - Nothing to share yet, but stay tuned

2026: Becoming a developer friendly eco-system

- Perf counters (Lukas, Adrian)
- Tracing/instrumentation (Mary, Lukas)
- Devcoredump (Daniel, Chia-I)

2026: New set of features

- Protected memory/content (Florent)
  - RFC posted
- Usermode queues (Mihail)
  - RFC posted
  - We need to have a functional shrinker first, to make things signalling/mem-reclaim safe

Behind the scenes

- Arm is working with Collabora and other partners to enable Panthor for all Mali CSF GPUs
- Currently only RK3xxx SoCs are widely available that can be tested with Panthor, but more are coming
- Panthor will receive support for new GPUs ahead of any other kernel driver (but some patches will be available downstream first)
- We are trying to move closer to a time where support for newer GPUs is present in mainline Linux before products are launched

Arm involvement with Panthor

- Early adopters/reviewers
- It is now considered to be the main kernel driver for future CSF-based GPUs
- We are working both on PanVK and on the kernel with Collabora to improve feature list and performance
- We have 2 maintainers (Steven Price and Liviu Dudau) connecting the internal teams and community
- All new GPU architectural changes are going to be considered and validated that they work on Panthor

New features we are working on

Module for sync with other devices

- We want to be able to use fences that are shared with drivers outside the DRM framework but use the Vulkan API
- sync_obj is a good API but support for it in Vulkan is still under MR
- Can we move sync_obj outside DRM?
- Can sync_obj be used by drivers outside DRM?
- Ultimate goal: have a kernel module that provides a generic sync fence API based on dma_fence and can be used by various drivers

Panthor Features we want to add

Job Submission:

- Publish a new version of the User Mode submission RFC

Protected jobs:

- Publish a new version of the Protected Mode RFC
    - The RFC relies on:
        - The use of the DMA Heap alloc/free API being called by the kernel driver.
            - Refs: https://lore.kernel.org/lkml/20230911023038.30649-1-yong.wu@mediatek.com/#t
        - The presence of a protected DMA Heap implementation:
            - Refs: https://lore.kernel.org/all/20250911135007.1275833-1-jens.wiklander@linaro.org/
    -

arm

Panthor Features we want to add

Power management:

- Future GPUs are supporting standalone DVFS/DCS OPP update
  - An RFC will be sent to present the Panthor governor driver:
    - drm/panthor/governor
  - Supported Features:
    - IPA power model, cooling device, GPU utilization reporting

Panthor Features we want to add

Memory management:

- Shrinker implementation RFC
    - We are evaluating how to implement the shrinker in Panthor
    - Considering 2 shrinkers:
        - One managing the:
            - Tiler Heap
        - Another for the rest of the memory:
            - User allocated memory imported via VM_BIND
            - Direction:
                - Transparent memory reclaim
                - May require `madvise` to further reduce memory footprint (prevent compressing scratch buffers)

dma_fence first commandment: you shall not allocate on-demand

- Mali GPUs rely on a grow-on-demand mechanism for their tiler memory
- Conflicts with the cross-device dma_fence contract: allocation in the dma-signalling path is forbidden
- We currently break this rule in at least 3 drivers: Lima, Panfrost and Panthor

Sparse GEM population goals

1) Allow non-blocking allocation so we can try to allocate in the dma-signalling path
2) Allow per-chunk population of the GEM object to avoid allocating memory we might never need
3) Provide common helpers to simplify driver's life and control the scope

Sparse GEM is not enough

- Non-blocking allocations can fail, what do we do then?
- Some times we have a HW fallback (incremental rendering on newer Mali GPUs)
  - Doesn’t work for more complex geometry pipelines (geometry/tessellation shaders)
- Have device-wide pre-populated memory pools to improve success rate of allocations in the dma-signalling path
- As a last resort, make the context unusable (DEVICE_LOST)
  - But try to report back why things failed so the UMD can allocate more memory upfront and hopefully not fail next time

Panthor Features we want to add

Instrumentation:

- Progress the Performance counter RFCs (Kernel and Mesa)
- Publish RFC to integrated Panthor ftrace events in Perfetto:
    - GPU activity trace
    - GPU mem total ftrace integration for Panthor
    - GPU frequency ftrace integration for Panthor

New GPU HW Features we want to add

- Publish Arm Mali G1-{Ultra/Premium/Pro} support
    - HW specific operations and register changes
        - Introduce GPU HW function operations and runtime abstraction
        - New HW interfaces will continue to rely on common Panthor feature
        - HW specific code will be abstracted

arm

Panthor Features we want to add

Power management:

- GPU always on power policy feature:
  - Integrates with runtimePM:

```
echo "on" > /sys/devices/platform/[a-f0-9]*.gpu/power/control
```

- Sleep On IDLE
  - Improve runtime GPU power usage by allowing the GPU to enter sleep mode when no job is executing (IDLE)

C/I and testing using FVP

- Arm is working on publishing Syzkaller definition for Panthor
  - The objective is to enable community to test security of Panthor
- Arm has worked on Panthor KUNIT tests
  - The objective is to enable regression testing on new patches
- Arm is providing Total Compute FVPs to validate Arm GPUs
  - https://developer.arm.com/Tools%20and%20Software/Fixed%20Virtual%20Platforms/Total%20Compute%20FVPs

Questions?

Thank you!

arm