Enhancements to the Raspberry Pi GPU driver stack

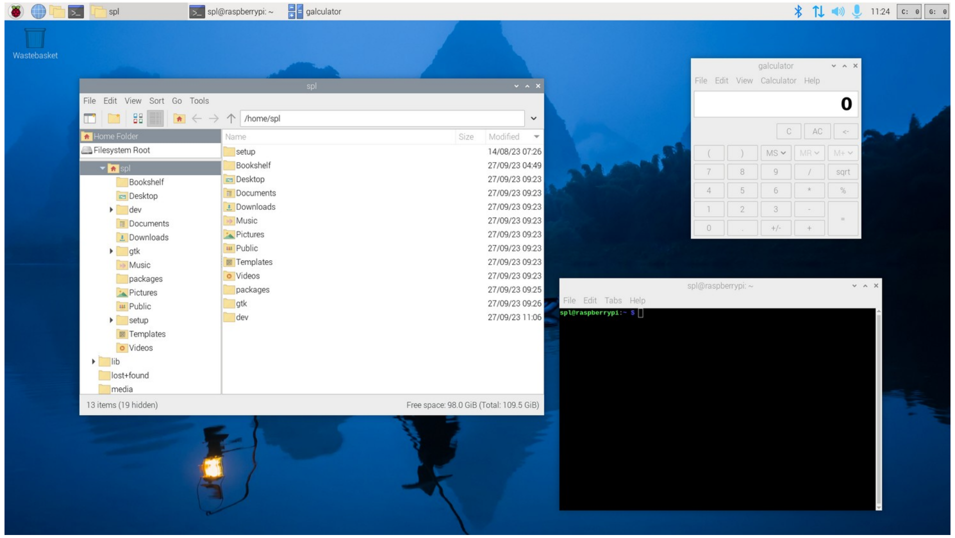
José María Casanova Crespo <jmcasanova@igalia.com>

XDC 2024 - Montreal

Who am I ?

- Chema Casanova (@txenoo@fosstodon.org)
- Open-Source developer at Igalia working at the Graphics Team.
- Working during the last years on the graphics stack of the Raspberry Pi devices, mainly on the Mesa user-space driver and optimizing the desktop experience.
- I’m representing the Igalia team working on Raspberry Pi GPU driver composed by Iago Toral, Alejandro Piñeiro, Juan Antonio Suárez, Maíra Canal and Christopher Michael.

One year ago at XDC 2023 A Coruña

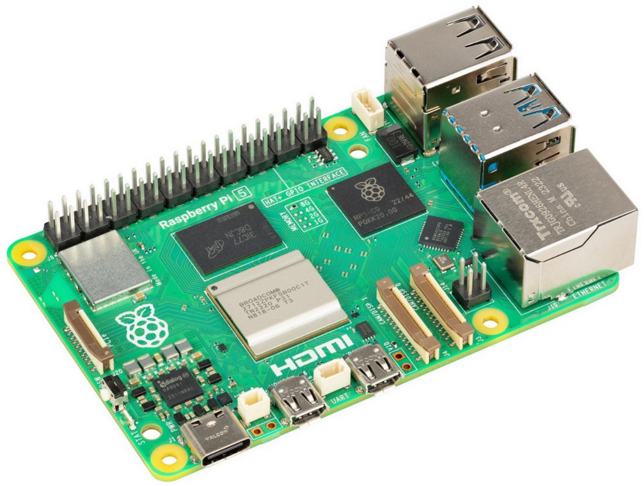


Name	Size	Modified
setup		14/08/23 07:26
Bookshelf		27/09/23 04:49
Desktop		27/09/23 09:23
Documents		27/09/23 09:23
Downloads		27/09/23 09:23
Music		27/09/23 09:23
Pictures		27/09/23 09:23
Public		27/09/23 09:23
Templates		27/09/23 09:23
Videos		27/09/23 09:23
packages		27/09/23 09:25
gtk		27/09/23 09:26
dev		27/09/23 11:06

13 items (19 hidden) Free space: 98.0 GiB (Total: 109.5 GiB)

```
spl@raspberrypi:~ $
```

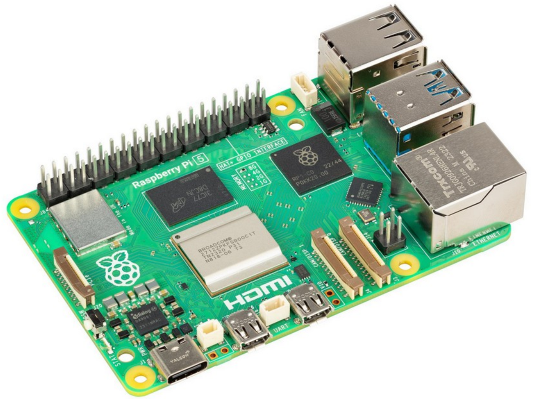
Bookworm Raspberry Pi OS (October 2023)



Raspberry Pi 5 Launched (October 2023)

Raspberry Pi 5

- GPU Broadcom V3D 7.1.7, same VideoCore architecture
- Higher clock rate than RPi4, up to 8 RTs, better support for subgroup operations, better instruction-level parallelism (but a bit more register pressure!), …
- Driver code merged into existing v3d and v3dv drivers in Mesa 23.3 and Linux Kernel 6.8.
- Same high-level feature support as Raspberry Pi 4:
  - Conformant OpenGL ES 3.1 and Vulkan 1.3.
  - Non-Conformant OpenGL 3.1



Raspberry Pi GPU driver stack

HW	GPU	Kernel Driver	Mesa Driver
Raspberry Pi 1-3	Broadcom VideoCore 4	vc4 (display+*render*)	*vc4 (GL/ES)*
Respberry Pi 4/5	Broadcom VideoCore 6/7	vc4 (display) *v3d (render)*	*v3d (GL/ES)* *v3dv (Vulkan)*

v3dv is Vulkan 1.3 Conformance

- v3dv achieved Vulkan 1.3 Conformance (August 2024)
- Exposed 18 new extensions
- Subgroups support:
  - Exposed new subgroups operation on Raspberrry Pi 5 because the ISA has better support for all that.

    VOTE, BALLOT, SHUFFLE, SHUFFLE_RELATIVE, QUAD
  - Enabled subgroups on the Fragment Stage on Raspberry Pi 4/5.
- Improved performance mainly in the shader compiler.
  - This improved the v3d OpenGL driver too.

From Vulkan 1.2 to 1.3

Exposed 18 new extensions

VK_EXT_depth_clamp_zero_one, VK_EXT_depth_clip_enable,
VK_EXT_extended_dynamic_state, VK_EXT_extended_dynamic_state2,
VK_EXT_multi_draw, VK_EXT_queue_family_foreign,
VK_EXT_shader_demote_to_helper_invocation, VK_EXT_subgroup_size_control,
VK_EXT_swapchain_maintenance1, **VK_KHR_dynamic_rendering**,
VK_KHR_index_type_uint8, VK_KHR_line_rasterization, VK_KHR_load_store_op_none,
VK_KHR_maintenance5, VK_KHR_shader_expect_assume,
VK_KHR_shader_relaxed_extended_instruction,
VK_KHR_shader_terminate_invocation, VK_KHR_vertex_attribute_divisor

OpenGL 3.1 & GLES 3.1

- New 7 new extensions available
- GL_ARB_depth_clamp, **GL_ARB_texture_barrier**, GL_EXT_EGL_image_storage_compression, GL_EXT_texture_storage, GL_EXT_color_buffer_half_float, GL_EXT_multi_draw_indirect, GL_EXT_shadow_samplers,
- Missing HW features, so OpenGL is not fully conformant
  - 8 Render Targets (supported in Raspberry Pi 5)
  - Non-seamless cubemap filtering
  - Required RGBA16_UNORM render format not supported

…but we can support everything else

Kernel: CPU jobs enabled

- Landed support for CPU jobs on v3d kernel driver.
  - For v3dv some Vulkan commands cannot be performed by the GPU alone: Timestamp queries, performance queries, indirect CSD jobs.
  - Moving CPU jobs to kernel space, allows DRM schedule to queue them, not stalling the submission, providing more efficient usage of the GPU.
- Available since Linux Kernel 6.8 and mesa 24.0.

Kernel: Super Pages capability

- The V3D MMU also supports 64KB and 1MB pages, called big and super pages, respectively.
- In order to create a big/super page, we need a contiguous memory region. That's why we use a separate mount point with THP (Transparent Huge Pages) enabled. In order to place the page table entries in the MMU, we iterate over the 16 4KB pages (for big pages) or 256 4KB pages (for super pages) and insert the PTE (Page Table Entry).
- Now userspace now can apply optimizations that are specific to kernels with Big/Super Pages support.

Kernel: Super Pages capability

- It shows an average performance improvement of 1.33% running GL and Vulkan traces.

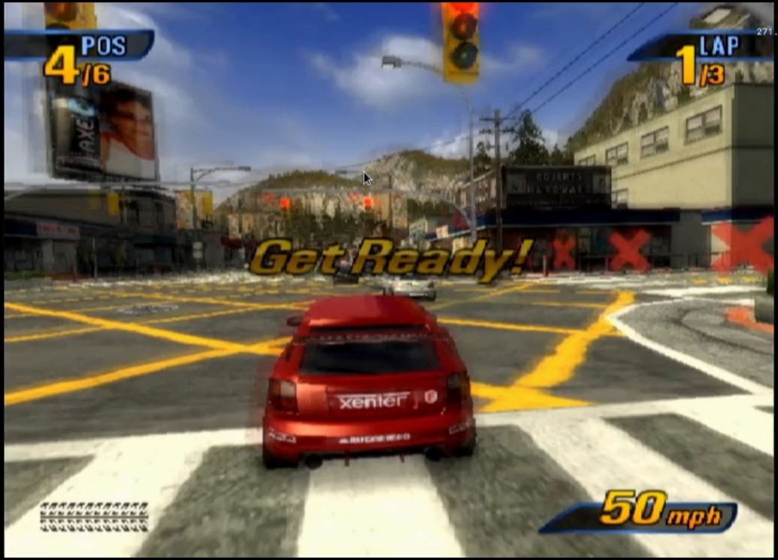
  total fps_avg in all runs: 1952.16 -> 1978.06 (1.33%)

  helped: 21

  HURT: 5
- But we get a **significant performance boost** in some emulation use cases.

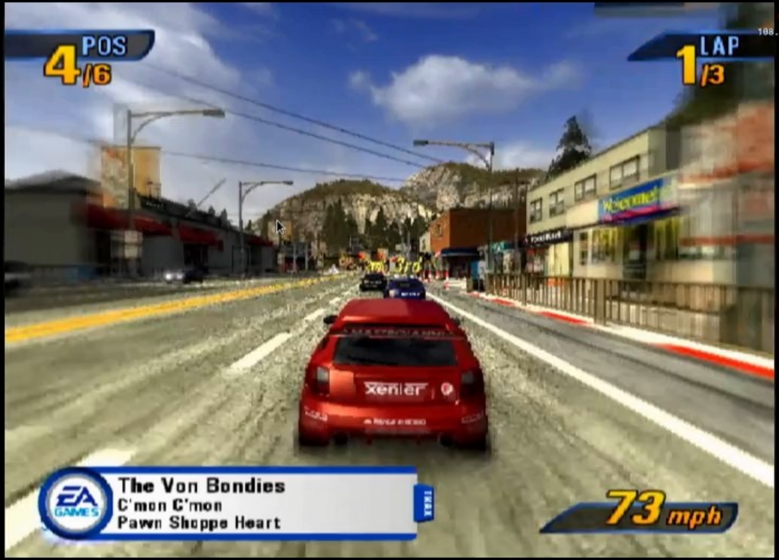
```
G: 3.68 [P] | V: 3.68
271.68ms (289.67ms worst)
      EE: 8.4% (22.79ms)
    GS: 99.5% (270.31ms)
      VU: 9.4% (25.57ms)
    GPU: 15.4% (41.90ms)
```

```
G: 9.22 [P] | V: 9.22
108.43ms (121.95ms worst)
     EE: 18.7% (20.24ms)
    GS: 98.9% (107.29ms)
     VU: 19.7% (21.38ms)
    GPU: 37.0% (40.07ms)
```



```
G: 3.68 [P] | V: 3.68
271.68ms (289.67ms worst)
EE: 8.4% (22.79ms)
GS: 99.5% (270.31ms)
VU: 9.4% (25.57ms)
GPU: 15.4% (41.90ms)
```

Without Super Pages



```
G: 9.22 [P] | V: 9.22
108.43ms (121.95ms worst)
EE: 18.7% (20.24ms)
GS: 98.9% (107.29ms)
VU: 19.7% (21.38ms)
GPU: 37.0% (40.07ms)
```

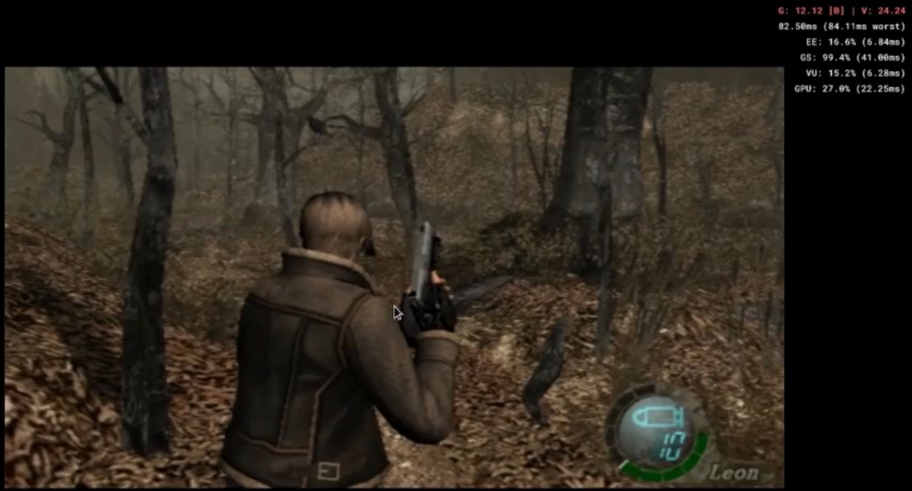
With Super Pages

Burnout 3: Takedown (PS2)

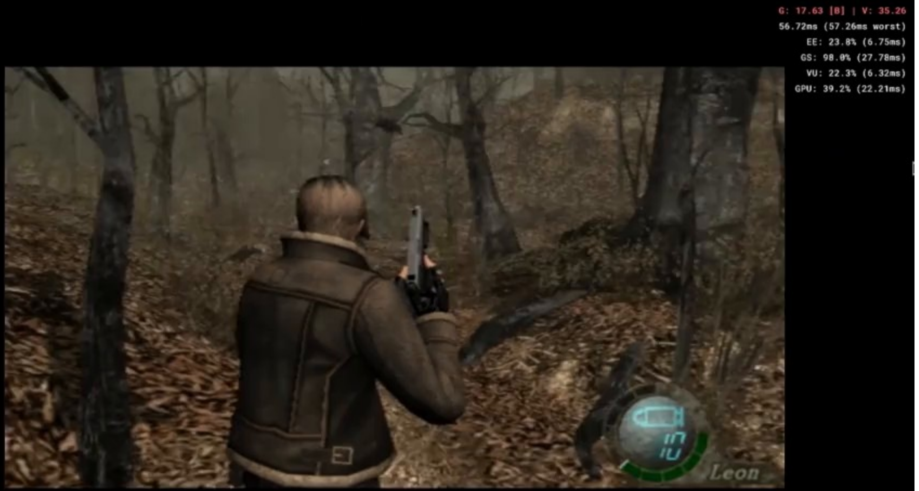
igalia

```
G: 12.12 [B] | V: 24.24
82.50ms (84.11ms worst)
EE: 16.6% (6.84ms)
GS: 99.4% (41.00ms)
VU: 15.2% (6.28ms)
GPU: 27.0% (22.25ms)
```

```
G: 17.63 [B] | V: 35.26
56.72ms (57.26ms worst)
EE: 23.8% (6.75ms)
GS: 98.0% (27.78ms)
VU: 22.3% (6.32ms)
GPU: 39.2% (22.21ms)
```



```
G: 12.12 [B] | V: 24.24
82.50ms (84.11ms worst)
EE: 16.6% (6.84ms)
GS: 99.4% (41.00ms)
VU: 15.2% (6.28ms)
GPU: 27.0% (22.25ms)
```



```
G: 17.63 [B] | V: 35.26
56.72ms (57.26ms worst)
EE: 23.8% (6.75ms)
GS: 98.0% (27.78ms)
VU: 22.3% (6.32ms)
GPU: 39.2% (22.21ms)
```

Without Super Pages

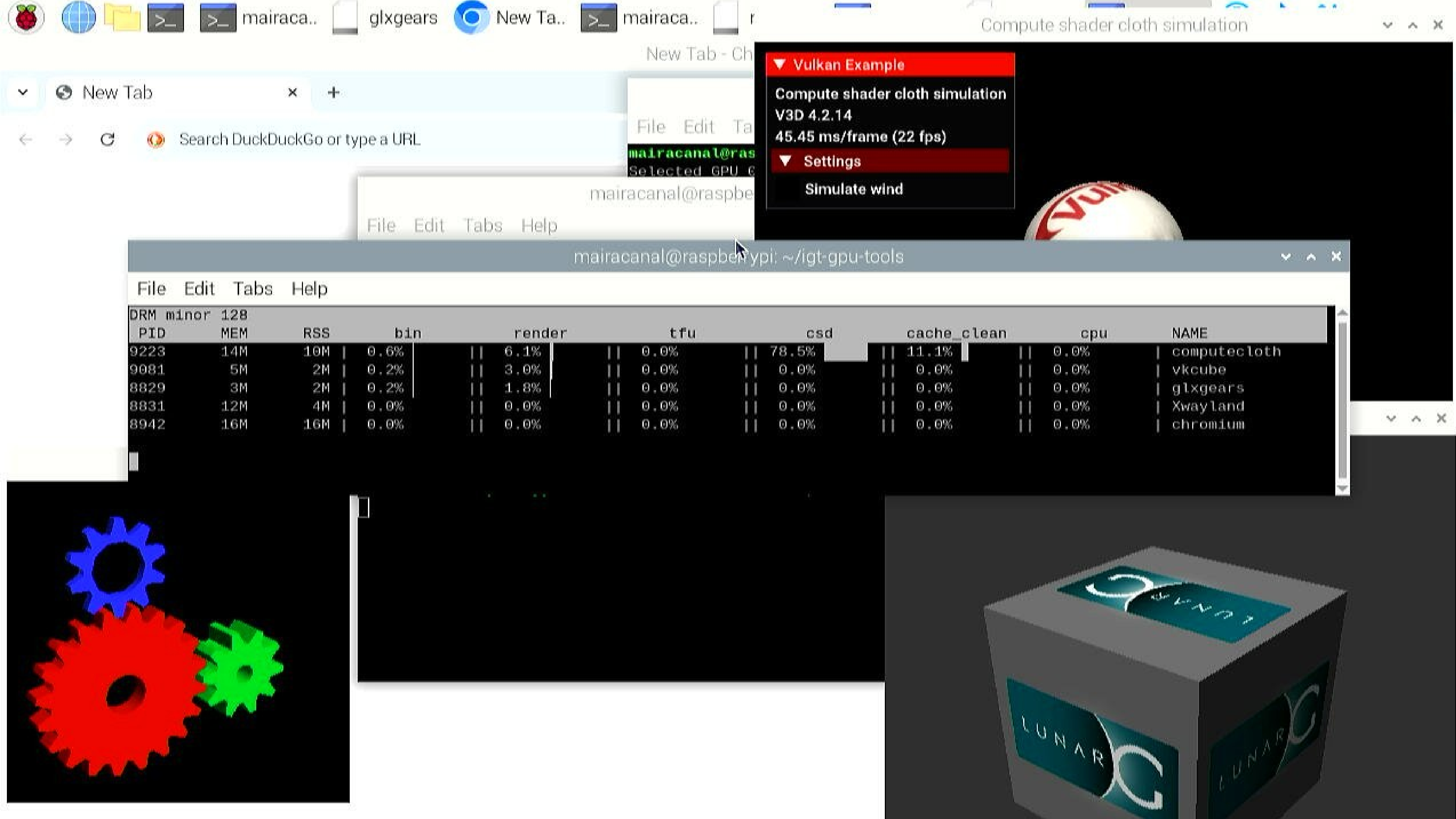
With Super Pages

Resident Evil 4 (PS2)

igalia

Kernel: Other improvements

- Moved performance counters definition from user to kernel space.
- Improved GPU usage stats per process exposing memory stats through fdinfo



Vulkan Example
Compute shader cloth simulation
V3D 4.2.14
45.45 ms/frame (22 fps)
Settings
Simulate wind

```
DRM minor 128
 PID      MEM      RSS      bin            render           tfu            csd         cache_clean        cpu        NAME
9223      14M      10M |  0.6%       ||   6.1%       ||   0.0%       ||  78.5%      ||  11.1%       ||   0.0%      | computecloth
9081       5M       2M |  0.2%       ||   3.0%       ||   0.0%       ||   0.0%      ||   0.0%       ||   0.0%      | vkcube
8829       3M       2M |  0.2%       ||   1.8%       ||   0.0%       ||   0.0%      ||   0.0%       ||   0.0%      | glxgears
8831      12M       4M |  0.0%       ||   0.0%       ||   0.0%       ||   0.0%      ||   0.0%       ||   0.0%      | Xwayland
8942      16M      16M |  0.0%       ||   0.0%       ||   0.0%       ||   0.0%      ||   0.0%       ||   0.0%      | chromium
```

Mesa: performance improvements

- We have recently focused on performance improvements on GPU limited scenarios using a target resolution of 1920x1080.

- We have analyzed the performance of V3D using several GLES gfxbench traces and we have achieved **an average of ~40% FPS improvement** in these scenarios.

One year of improvements

Bookworm Raspberry OS Mesa 23.2.1 vs Main f3f4f0fb8342a7 (October 7th 2024)

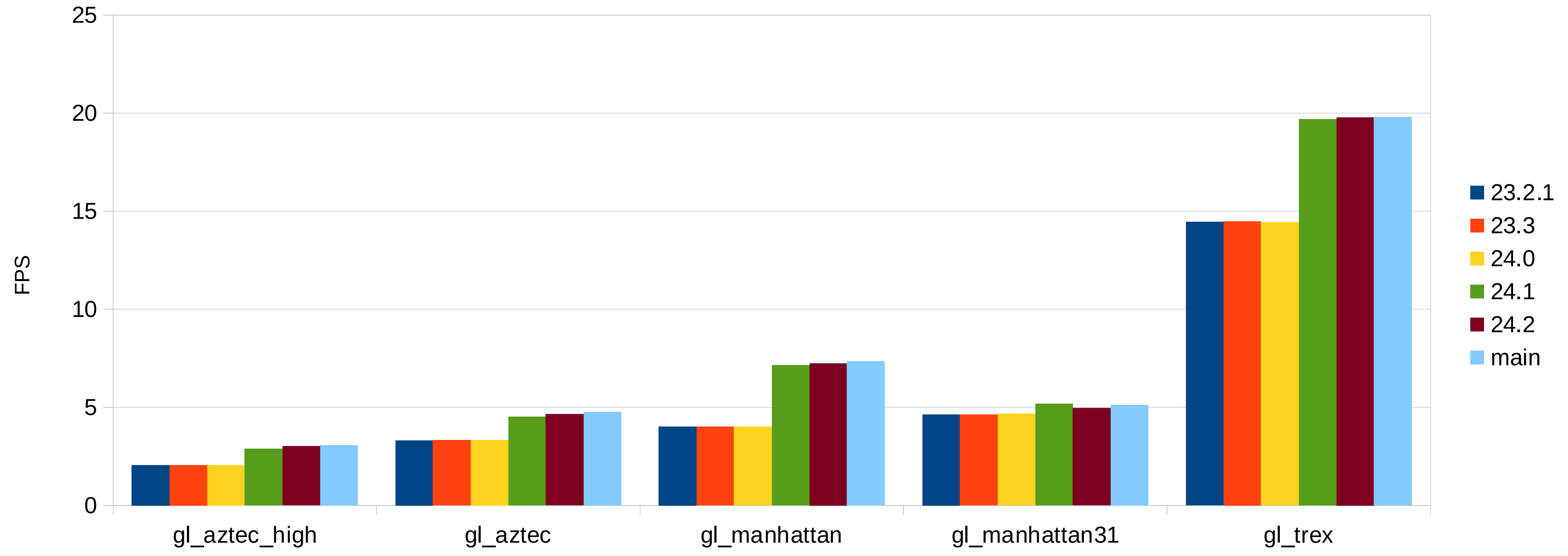
fps_avg	helped:	gl_manhattan.trace:	4.03 -> 7.36 (82.78%)
fps_avg	helped:	gl_aztec_high.trace:	2.06 -> 3.07 (49.12%)
fps_avg	helped:	gl_aztec.trace:	3.31 -> 4.76 (43.85%)
fps_avg	helped:	gl_trex.trace:	14.45 -> 19.80 (37.01%)
fps_avg	helped:	gl_manhattan31.trace:	4.64 -> 5.12 (10.41%)

total fps_avg in all runs: 28.48 -> 40.10 **(40.82%)**

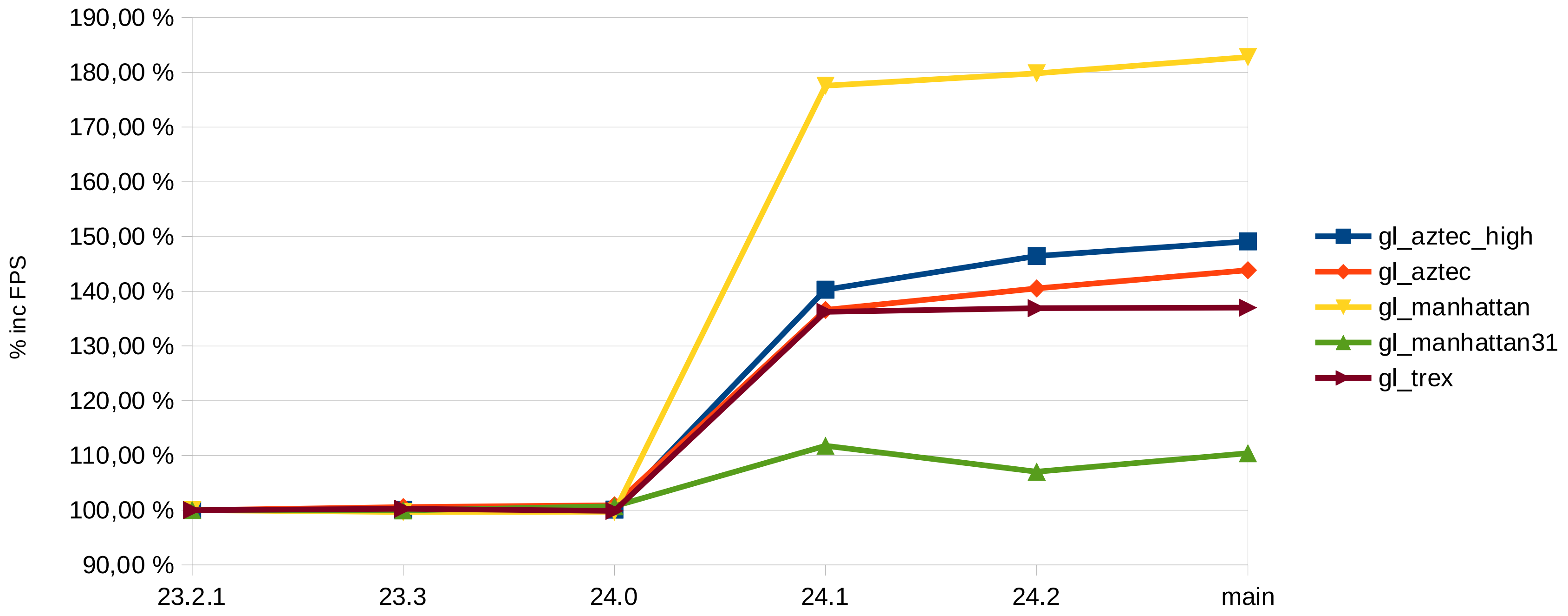
helped: 5

HURT: 0

Performance over history



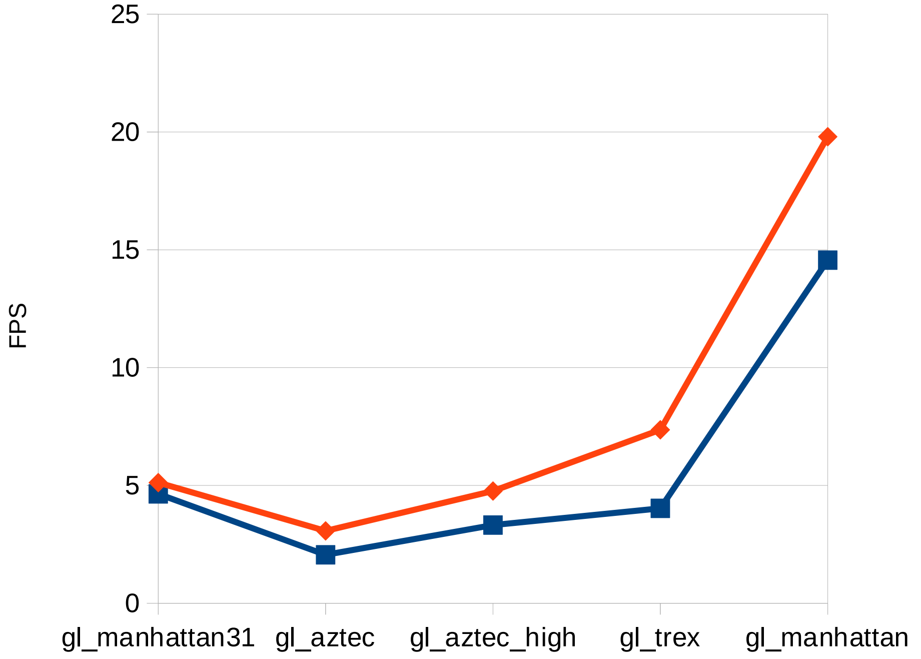
Performance Improvement

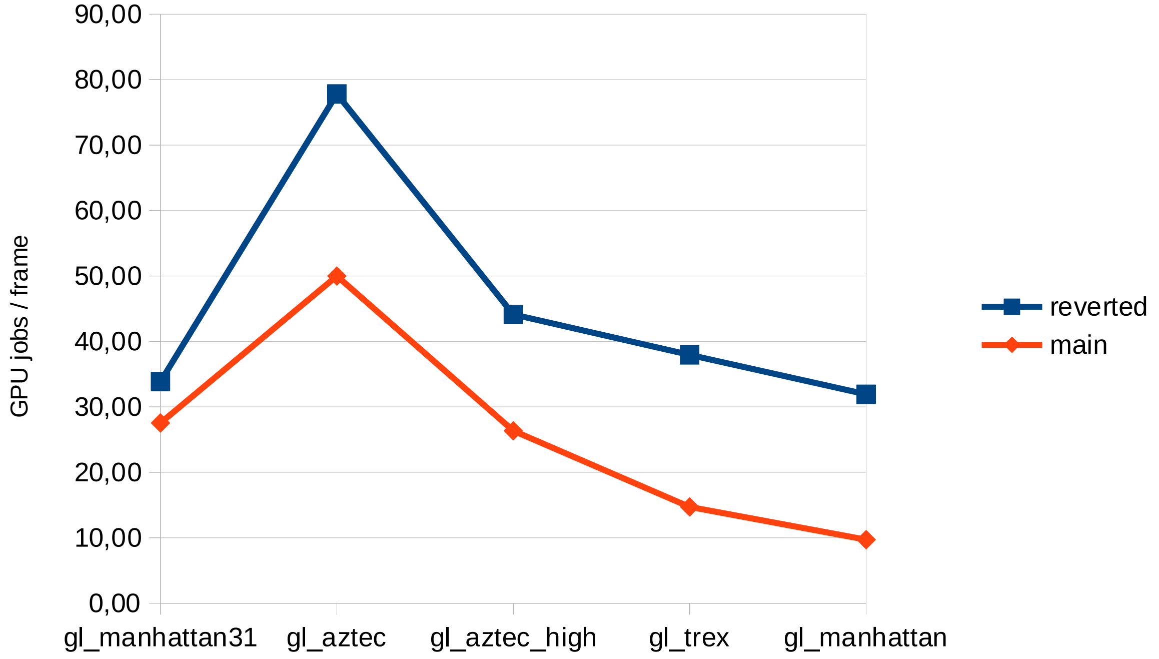


Reducing the number of flushes

- We identified that v3d was being too conservative during the implementation of **ARB_texture_barrier** as the driver passed all the tests with an empty implementation.
- v3d was flushing jobs that wrote a resource that was going to be sampled.
- But there is no need in the cases where the job reading the resource is the same one that that was writing on it, as updates are available in the cache.
- So merging draw calls in the same GPU jobs avoids extra loads/stores of the tile buffer and provides a **significant performance improvement.**

Reducing the number of flushes





Reducing the number of flushes

```
fps_avg  helped:  gl_manhattan.trace:                    4.03 ->  7.36 (82.77%)
fps_avg  helped:  gl_aztec_high.trace:                   2.06 ->  3.07 (49.36%)
fps_avg  helped:  gl_aztec.trace:                        3.32 ->  4.76 (43.57%)
fps_avg  helped:  gl_trex.trace:                        14.56 -> 19.80 (35.97%)
fps_avg  helped:  gl_manhattan31.trace:                  4.64 ->  5.12 (10.30%)
```

total fps_avg in all runs: 28.60 -> 40.10 (**40.24%**)

helped: 5

HURT: 0

Compiler backend optimizations

We have implemented multiple compiler optimizations reducing the total number of instructions more than 4%

```
total instructions in shared programs: 630354 -> 604028 (-4.18%)
instructions in affected programs: 572837 -> 546511 (-4.60%)
helped: 8072
HURT: 995
```

broadcom/compiler: perf changes

- generate mali opcodes for clamping on Pi5 (inst -0.11%)
- don't use small immediates in geometry stages (inst -0.18%)
- don't add const offset to unifa if it is 0
- avoid register conflict with ldunif(a) and ldvary (inst -0.01%)
- skip small immediates optimization on vpm instructions (inst -2.46%)
- emit instructions producing flags earlier (spills -0.33%)
- don't read excess channels on image loads (inst -4.64% on vk-cts subgroup)
- fix SFU check for 7.1 (sfu stalls -6.36%)
- update image store lowering to use v71 new packing instructions (-7.63% on vk-cts subgroup)

Next steps

- Continue working on performance improvements on V3D and V3DV.
  - Continue working on improving the compiler backend.
  - Work on reducing the load/store operations.
  - Kernel space driver performance.
  - Integrate with Perfetto for GPU performance monitoring
- Work on implementing more Vulkan/OpenGL extensions.

Questions ?

Join us!

https://www.igalia.com/jobs/

Enhancements to the Raspberry Pi GPU driver stack.

José María Casanova Crespo <jmcasanova@igalia.com>

XDC 2024 – Montreal