The state of video offloading on the Linux Desktop

About convergence and longer battery life

FOSDEM'24

COLLABORA

Content

- What am I even talking about?
- Current status and recent developments
- Demos and benchmarks
- Notes
- Questions?

What am I even talking about?

What am I even talking about?

- Hardware video decoders (VA-API, V4L2, Vulkan) decode to **YCbCr**/YUV formats
- **Display controllers** often can convert to **RGB**, scale and rotate
- Needed to get the maximum out of your hardware

What am I even talking about?

- Embedded: apps can use KMS/DRM directly
  - Gstreamer: kmssink
  - MPV: --vo=drm
  - QT: QT_QPA_PLATFORM=eglfs
  - Kodi: --windowing=gbm
  - Various proprietary solutions
- X11: Xvideo
  - Never really took off

What am I even talking about?

What am I even talking about?

- Wayland:
  - First publicly demonstrated video offloading in 2014
  - Gstreamer: waylandsink / gtkwaylandsink
  - MPV: --vo=dmabuf-wayland
  - Limited compositor and client support, practically constrained to embedded

What am I even talking about?

- Wayland matured on the desktop
- Apps/toolkits started to use GL/Vulkan
- Atomic-KMS:
  - DRM_MODE_ATOMIC_TEST_ONLY
- DMABuf:
  - Explicit modifiers (tiling/compression)

Current state and recent developments

Current state and recent developments

- Mutter / Gnome-Shell:
  - YCbCr support (Gnome 45)
  - KMS scaling support (Gnome 46 🤞)
- GTK4:
  - GtkGraphicsOffload (GTK 4.14)
  - Light Video / livi (🤞)
  - Gtk4paintablesink (🤞)

Current state and recent developments

Current state and recent developments

Current state and recent developments

- ChromeOS
  - Is witching to Wayland (Lacros/Exo)
  - Experimental patches for non-ChromeOS Wayland backend exist
  - --enable-hardware-overlays=single-fullscreen,single-on-top,underlay
  - Sommelier lacking behind (Linux apps in containers)

Current state and recent developments

- Compositors:
  - Weston: Good hardware plane support
  - Mutter: Fullscreen only
  - Kwin: NV12 support landed for Plasma 6.0 / Fullscreen only
  - wlroots:
    - Sway: scene graph API support landed
    - Wayfire, Hyprland: not using wlroots scene graph / render API

Current state and recent developments

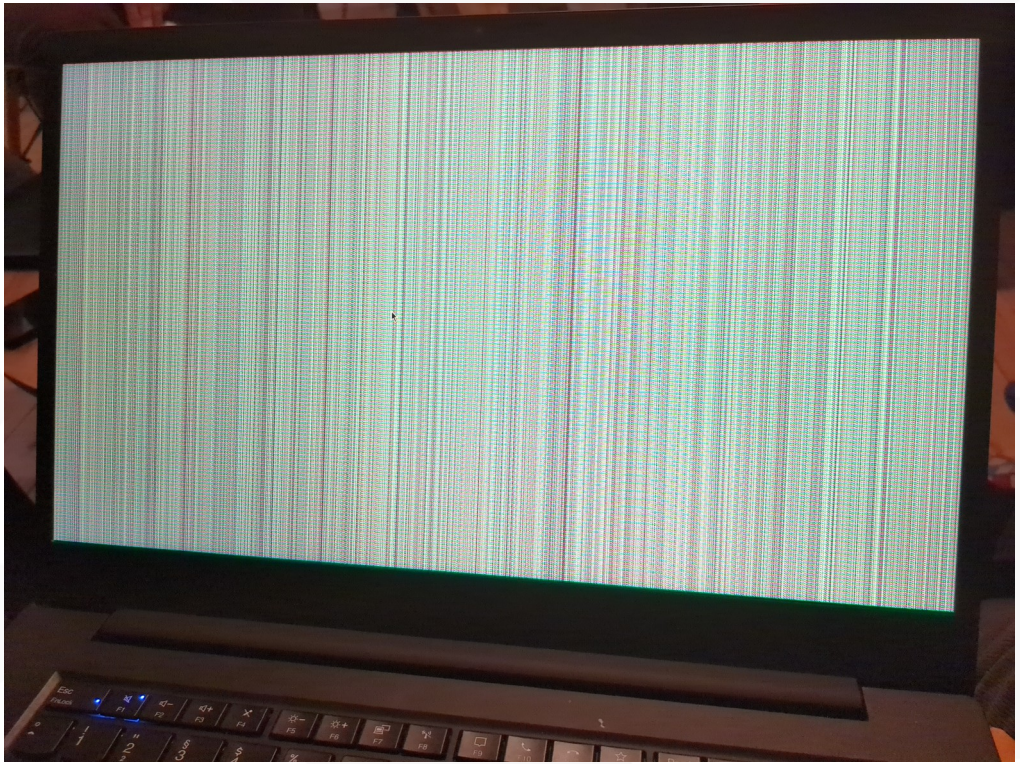
- Missing Wayland protocols:
  - Color representation protocol (10bit content/BT.2020)
  - Color management protocol (HDR)
  - 2024?

Current state and recent developments

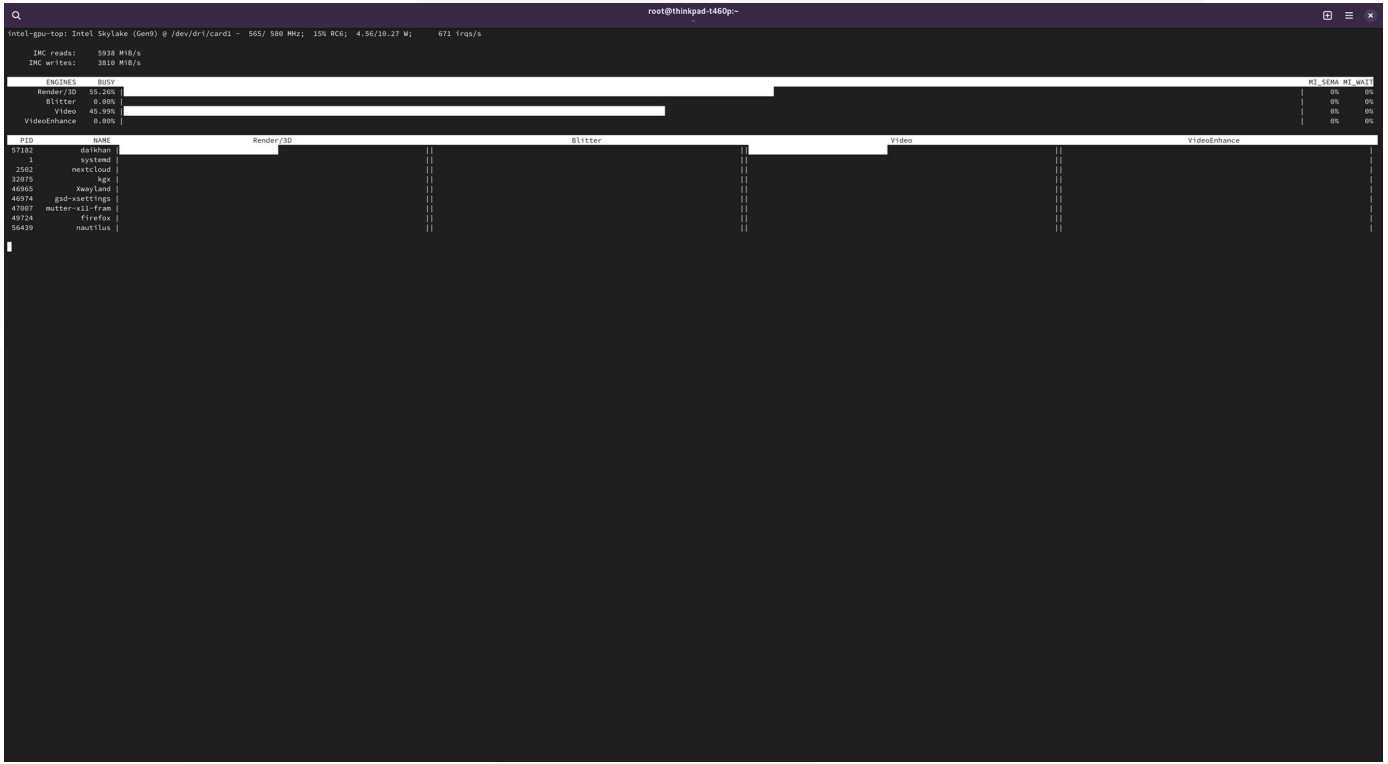
- Most compositors have at least basic support for YCbCr formats and fullscreen playback
- Lots of positive dynamics
- Some features still missing
- Good moment to join the effort!

Demos and benchmarks

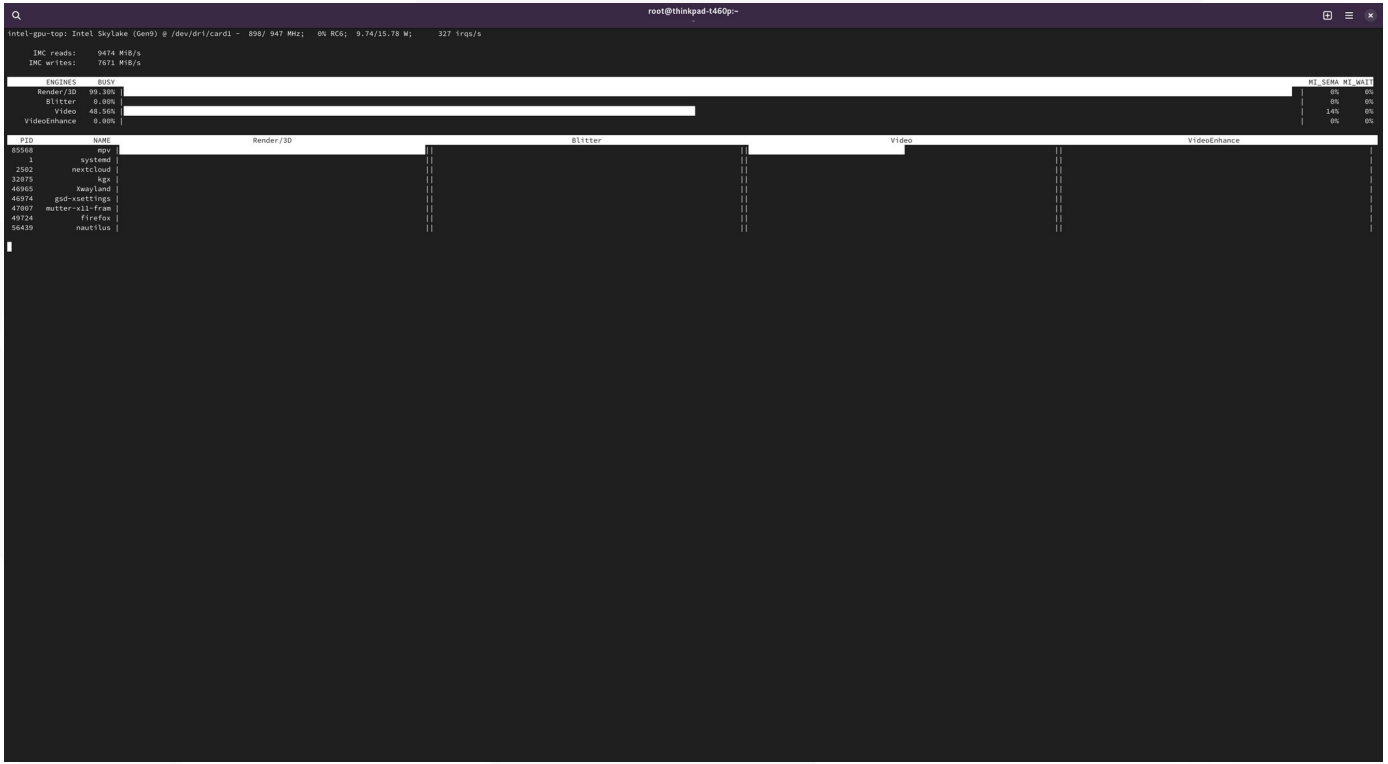
Demos and benchmarks



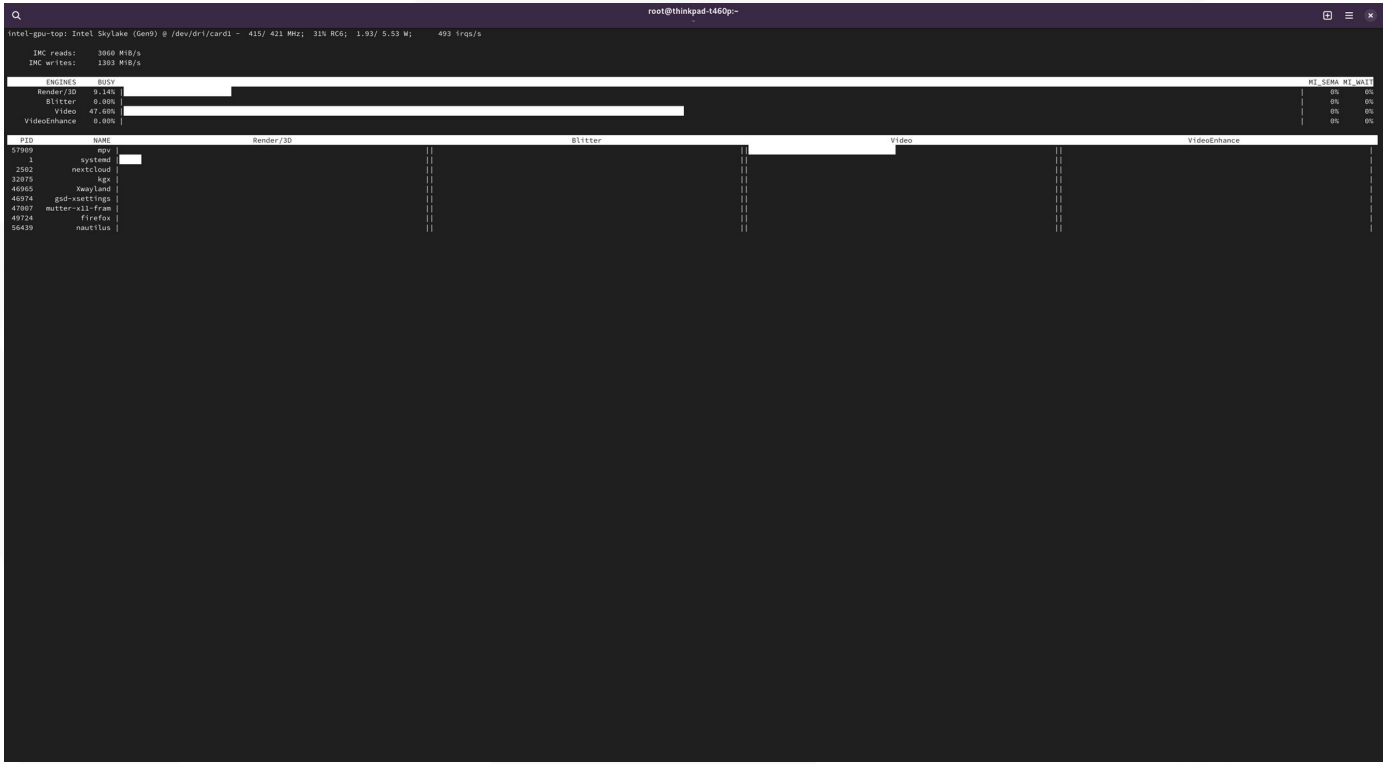
Demos and benchmarks



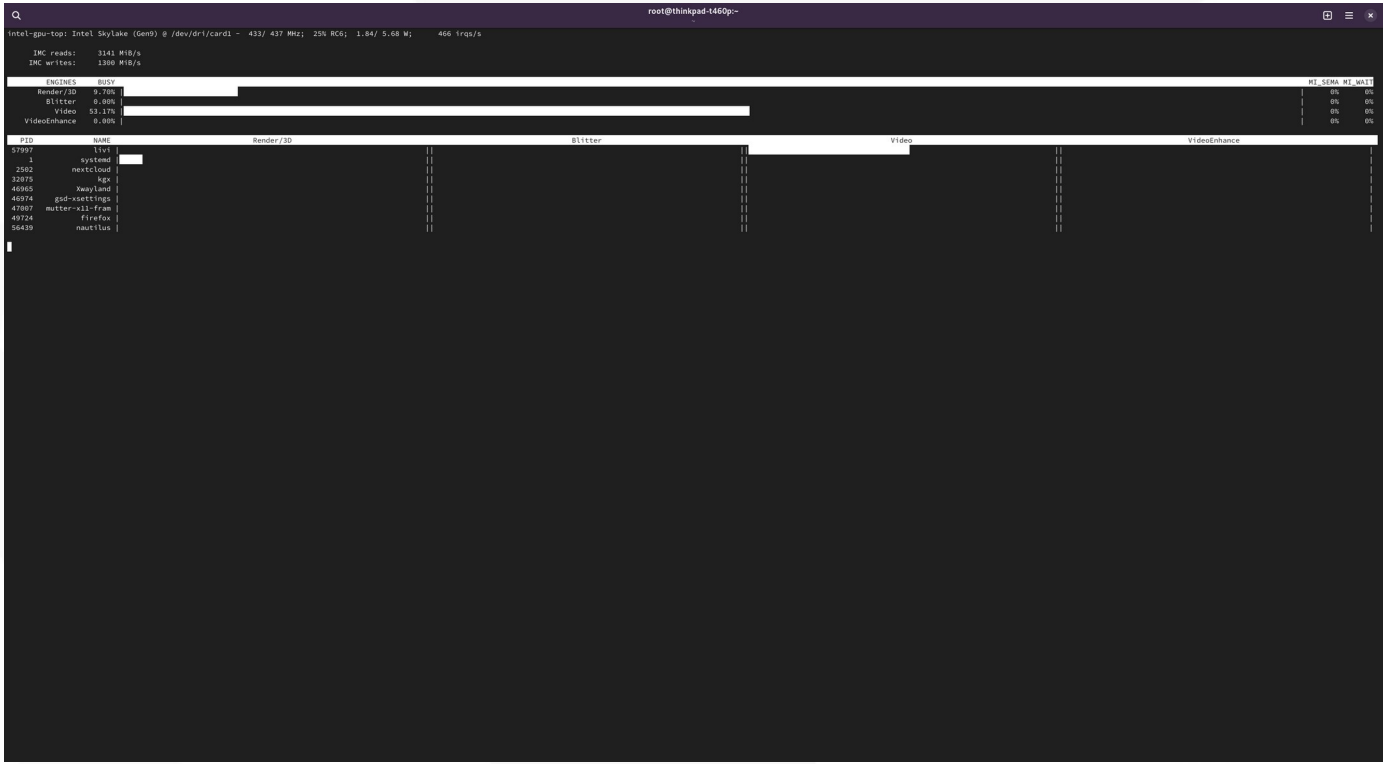
Demos and benchmarks



Demos and benchmarks



Demos and benchmarks



Demos and benchmarks

- Current GTK4 players: > 10W
- MPV default: > 15W
- MPV –vo=dmabuf-wayland: ~5.5W
- Livi / upcoming GTK4 players: ~5.5W

Demos and benchmarks

- Offloading has significant resource consumption advantages on most hardware
- Complex toolkits and apps can be best in class
- Linux Desktop users help with finding bugs and improving quality!

Notes

Notes

- DRM. That other one. **Digital Rights Management**.
- Technically becomes easy: just another modifier
- When not possible: place-holder

Notes

- Experimental patches for V4L2 stateless:
  - Gstreamer
  - Chromium
- Experimental patches for V4L2 stateful:
  - Gstreamer (Raspberry Pi 4)

Notes

- Upcoming hardware will benefit both **more and less** than current generations
  - More powerful/efficient hardware → less impact
  - More independent hardware blocks → turn off parts completely → more impact

Notes

- Mesa:
  - OES_EGL_image_external only on GLES → Gstreamer + GTK4 switched
  - OpenGL only inofficially supported
  - Vulkan will make things better
- Nvidia
  - 550.40.07: “Added support for R8, GR88 and YCbCr GBM formats.”

Questions?

Thank you!

Contact

- https://www.collabora.com/news-and-blog/
- Follow me on Mastodon: https://floss.social/@rmader

Sources

- https://en.wikipedia.org/wiki/YCbCr#/media/File:CCD.png
- https://www.youtube.com/watch?v=GtXQJ0c5q0k
- https://blog.gtk.org/2023/11/15/introducing-graphics-offload/