cerno

Towards a More Reliable Display Stack

X.org Developers Conference - Maxime Ripard

17/09/2021 - Virtual

Testing in KMS

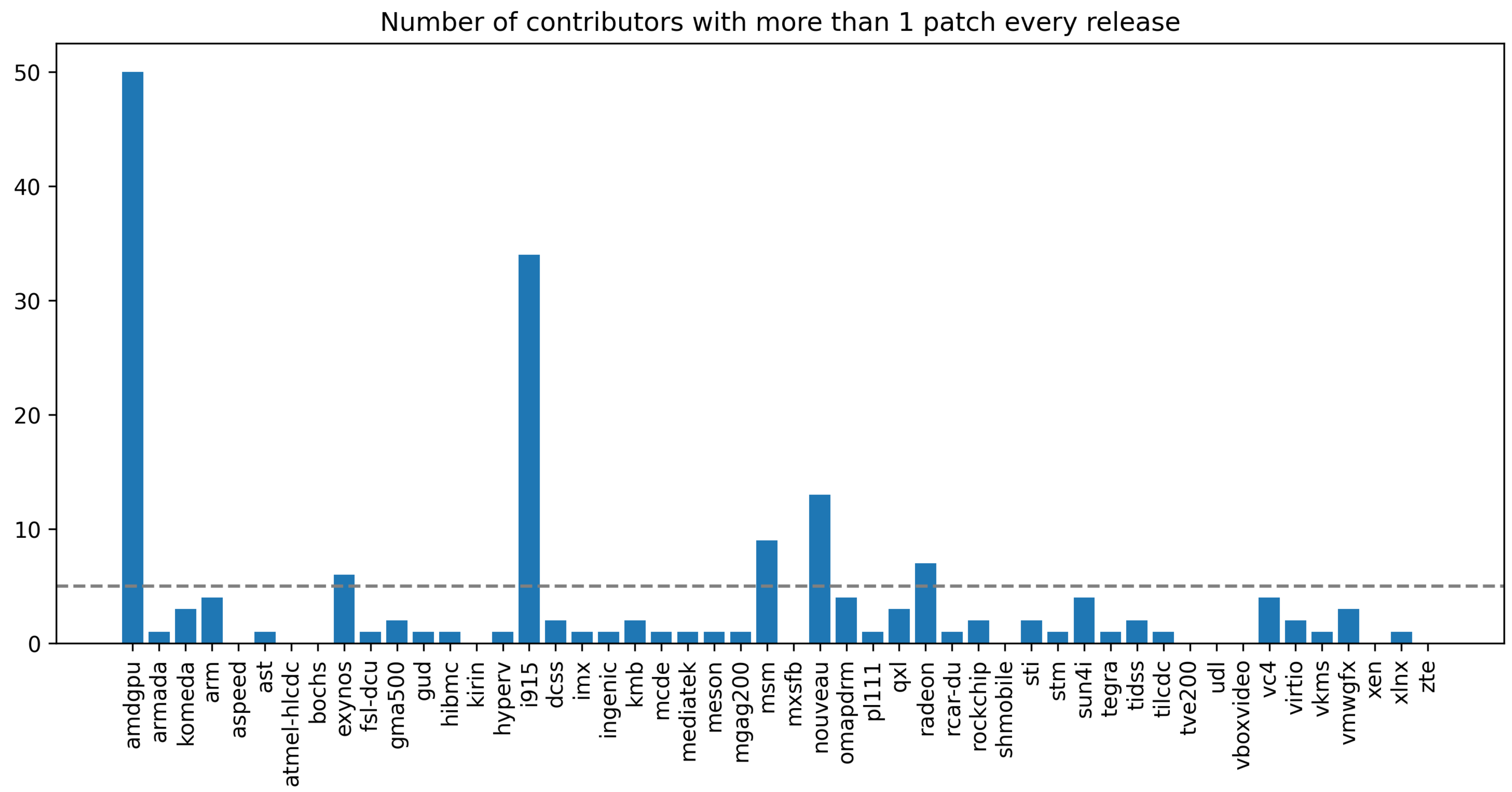
KMS Succeeded

- Thanks to the massive effort to make DRM/KMS easier, more than 60 KMS Drivers in tree (and counting)
- 1500-2000 patches to `drivers/gpu/drm` each release
- `fbdev` is now pretty much dead, and only the uAPI is still (slightly) relevant
- It's now the de-facto standard, with all the drivers and display-related features targeting KMS.
- And on a maintenance front, the use of helpers in most drivers make it fairly easy to maintain as well given its size and importance.

Drivers are mostly a part-time effort

- The amount of features and cases implemented by the core is massive
- ... And it's easy to overlook or under-estimate some of them ...
- ... Or to misunderstand their requirements or side-effects ...
- ... Or to be unable to test them easily.

KMS Drivers Maintenance



Hardware is barely accessible

- Controllers might not be easy to access (proprietary, confidential, not produced anymore, etc.)
- Or the one available might not expose all of the hardware features
- So we end up in a situation where people with hardware don't have the knowledge and people with the knowledge don't have the hardware

A List of Bugs

- Scrambling Support Addition: #4302
- Short hotplug pulse aren't detected: #4313
- CPU crash with CEC access while disabled: #4319
- TV remains black out of standby in 4k: #4411
- TV remains black out of standby in 4k, with Kodi: #4486
- Deadlock when waking up a TV with CEC: #4553

CI

- These issues are fairly standard behaviour
- Should be reported by CI
- FreeDesktop GitLab
  - Being discussed, but not a thing yet
- KernelCI
  - No display tests
- More importantly, which tools should we run?

IGT GPU Tools

- An extensive test suite, with tests for both the display and rendering sides, and for both generic and vendor-specific features
- The policy to test every new feature through an IGT test is great
- ~2000 Tests
- Well maintained
- Tests and documents everything

IGT Issues: Deployment

- Fairly big number of dependencies (and big ones)
- Pretty much requires a "real" distribution
- Cross-compilation is fairly hard too
- Docker helps marginally:
  - We can't always have Docker / podman
  - A build of IGT with Dockerfile.build-debian-minimal takes around 700MB

Typical Embedded Device

- An Embedded Device has:
  - One (weak) CPU
  - 64MB of RAM or more
  - A discrete GPU (sometimes)
  - Around 128MB of Flash Storage
  - No Network
  - No HDMI, DisplayPort but rather MIPI-DPI, MIPI-DSI or LVDS

IGT Issues: Test Suites

- Huge Number of Tests, and support for tests suites
- Only 3 users: intel, vc4 and v3d
- Hard to see which tests we want to run on a given platform (and we can't run all of them)
- Some tests are long, and not ideal for CI
- No suite that any driver must pass

IGT Issues: Features

- Mostly tests the user-space API and the driver behaviour
- Vkms allows to test the core in depth
- Writeback allows to test only some parts of the driver
- It's really more of a test to see if the driver reports an error and behaves properly, not if the result is actually valid

IGT: Chamelium?

- Device made by Google for ChromeOS
- Allows to capture and retrieve frames and their CRC through the network
- But:
  - Expensive, difficult to source
  - Limited number of input (HDMI, DisplayPort, VGA)
  - Requires a network connection
  - Limited testing and difficult to extend (VHDL)

IGT Issues: Blind Spots

- Difficult to setup for part-time developers
- Impossible to deploy on some platforms or devices
- The driver might have no way to tell that the hardware doesn't output anything (HDMI SCDC, Unidirectional busses, etc.)

Ideal World

- We'd need a tool that:
  - Can be deployed easily on any platform (supported by KMS)
    - Overall size in 10MB order of magnitude
    - Can run without network
    - Can be cross-compiled
  - No ramp-up time or internal knowledge of the tool needed
  - Can test the driver output, with cheap hardware

A Possible Solution

The Plan™

- IGT is the full test-suite and we definitely need to keep it
- But need to write a tool with:
  - Be easy to deploy, on any platform of any architecture (using KMS)
  - All the KMS drivers can pass all the tests (à la v4l2-compliance)
  - Can test the display output, using relatively cheap (~100$) hardware, and without network
  - Can test multiple interfaces, including "internal" ones

Architecture

- Three Components:
  - A tool that runs on the device under test
  - An optional board to capture the DUT output
  - A tool that runs on that board and processes the captured frames

DUT

- Rust Application
  - Statically compiled
  - Dependency only on the C library
- Atomic KMS Application
- Runs all the local tests on the device
- 4MB

Prototype

- Based on an HDMI to MIPI-CSI Bridge
- Available to most (but MIPI-DSI?) interfaces
- MIPI-CSI Capture pretty ubiquitous too
- Prototype based on the RaspberryPi3 and Pi4 and Toshiba TC358743XBG Bridge

Capture

- Rust V4L2 Application
- Runs a (configurable) test scenario
- Sets up the bridge and capture interface, sets the EDID
- Validates the captured frames

Frame Validation

- Every frame sent by the DUT contains a header
- This header contains a counter and a hash
- Validates that the frames are in order, and that the hash is correct
- Takes 3-7ms to process a 1920x1080 frame

Limitations

- Rely on interrupt-based hotplug detection to switch resolutions, will not work with poll-based devices
- Validation based on a hash is fragile and will not be able to test some features (like colorspace conversions)
- We don't have a way to send parameters to the DUT
- MIPI-CSI bridges and capture devices for 4k resolutions are rare

Additional Features

- Integration into a CI environment
- Infoframes
- 4k
- Audio Support
- CEC Support
- Other Interfaces

Any Questions?

Contact

Maxime Ripard

https://www.cerno.tech - maxime@cerno.tech

in/mripard