VLC audio/video outputs

Linux.conf.au Multimedia & Music

Rémi Denis-Courmont

VideoLAN project

Auckland, 12th January 2015

Outline

Pipeline

Audio output

Video output

Note well

The opinions hereby expressed represent the personal views of the author.

Attendees advisory

- I speak fast.
- I may not articulate adequately.

Attendees advisory

- I speak fast.
- I may not articulate adequately.

Do interrupt me if needed.

Outline

Pipeline

Audio output

Video output

Pipeline

- defacto standard among multimedia frameworks
- enforced by the specifications

Pipeline overview

- byte stream reader
- format parser
- (packetizers)
- decoders: audio, video, text

Pipeline overview

- byte stream reader
- format parser
- (packetizers)
- decoders: audio, video, text
- filters
- blending / overlay
- outputs

Driving the pipeline

- buffers here and there
- rate control, drift compensation and lip sync

Outline

Pipeline

Audio output

Video output

Buffers

Media playback has audibly long buffers

- avoids underruns due to scheduling: no stutter

Buffers

Media playback has audibly long buffers

- avoids underruns due to scheduling:
  no stutter
- reduces (or even eliminates) periodic interrupts:
  lower power

Buffers

Media playback has audibly long buffers

- avoids underruns due to scheduling:
  no stutter
- reduces (or even eliminates) periodic interrupts:
  lower power
- unlike games and UIs:
  needs special support to avoid latency and glitches

Buffer requirements

- Play-out latency estimate
  - Maintain lip synchronization
  - Control upstream pipeline rate
- Drain or fill levels estimate:
  - Normal EOS without loosing last samples

Interactive requirements

- Flush: user stop or exit
- Pause/resume
  (some APIs cycle their playout buffer)
- Volume and mute control
  - interactive volume control
  - per stream (not whole device!)

Misc. requirements

- Device enumeration, hotplug events
- Configuration and format negotiation

Common problems

- confusing total latency and buffer usage (or no timing infos whatsoever),
- no (glitch-free) pause/resume
- no explicit drain and/or flush operations
- device-wide volume controls only
- no channels layout
- missing or broken device management

JACK

- specific constraints:
  - low latency
  - manually routing
  - always single precision
- works around most of the requirements
- not adequate for general use

ALSA - channels

- channels count
  - hardware cap, not physical speakers setup
  - *plug* plugin drops extra channels silently

ALSA - channels

- channels count
  - hardware cap, not physical speakers setup
  - *plug* plugin drops extra channels silently
- channels map
  - not until recently (and not all drivers)
  - also not necessarily wired speakers
- SW defaults to stereo with per-app knobs
- digital output: similar problems

ALSA (cont'd)

- no stream volume
  (and HW volume controls a big unabstracted mess)

ALSA (cont'd)

- no stream volume
  (and HW volume controls a big unabstracted mess)
- defective device management:
  - no hot plug/unplug events (and no udev integration)
  - confused channels and outputs

OSS

- Questionable API design (*ioctl*)
- Most outstanding functional issues fixed in version 4.
- Mostly dead: last version 4.2 in 2010

sndio

- Not Invented Here syndrome from OpenBSD
- also RoarAudio server on Linux (almost dead)
- each and every possible mistake

sndio

- Not Invented Here syndrome from OpenBSD
- also RoarAudio server on Linux (almost dead)
- each and every possible mistake
- OK, except per-stream volume

PulseAudio

- Decent and well documented
  (seems to borrow from Windows Vista)
- Some bugs, maintainance handed over poorly
- Bonuses: live fail-over, stream meta infos

Overall

- Low-latency and manual setup: JACK
- Embedded: ALSA (without plugins)
- All else: PulseAudio (I wish)

Outline

Pipeline

Audio output

Video output

Requirements

- YCbCr colour space
- colour subsampling
- more than 8-bits per component (very near future)
- planar picture formar
- scaling
- blending (subs, overlay)

Goodies

- filtering
    - deinterlacing
    - gamma correction
    - noise reduction (nice to have)
- hardware decoding acceleration and pass-through?

Video output with X11

- base (+ MIT-SHM)
- XVideo extension (+ MIT-SHM)
- GLX extension
- Render extension
- VDPAU-X11
- VA-X11
- EGL-X11

Video output with Wayland

- base
- ~~XVideo~~
- (Wayland-EGL)
- Wayland scaler (wl_scaler)
- ~~VDPAU~~
- VA-Wayland
- Wayland-EGL

XVideo extension

- originally meant for dedicated hardware overlay
- no compositing, no blending
    - except with pixmaps support
- inconsistent cropping
- provided for backward compability
- overdue for deprecation

Renger extension

- roughly equivalent to wl_scaler (but more boilerplate)
- only RGB
- only 8-bits per component

DRM

- X11, Wayland, headless
- hardware-dependant
    - not provided by, err, some drivers
- intended for GL (and VA), not for applications (say Wayland/Weston developers)

VDPAU & VA

- not vendor-neutral
  - VDPAU: AMD(Mesa), NVIDIA, Nouveau(Mesa)
  - VA: Intel OSC
  - XvBA: AMD(Catalyst)
- high-depth coming?
- vvvv ??

OpenGL

- versions and extensions hell
- shaders for colour space convesion
    - supports high-depth
- code reuse on other platforms
- interoperable with VDPAU

Overall

- VDPAU or VA where applicable especially hardware-accelerated decoding
- OpenGL
- buggy drivers...

Thanks to VideoLAN for sponsporing most of the costs.

Any questions?