FFmpeg 6.0 - dav1d 1.1.0
VLC.js

Jean-Baptiste Kempf

President of VideoLAN

Active member in FFmpeg, x264, dav1d and VLC communities

FFmpeg @ FOSDEM

Not enough people speaks about FFmpeg

- FOSDEM & other events
- Hacker News *& Voici*

Many things are happening

- Forks are long gone
- Community health

FFmpeg 5.0 "Lorentz" (winter 2022)

Release

- 150 contributors, ~2.5k files changed, ~200kLoC touched
- Long to come, matching New release schedule
- Major version bump - many obsolete APIs removed

Changes

- Avcodec encoding+decoding change
  - Audio & Video single API
  - Decoupled codec input & output
  - Encoders to output data into user-managed buffers
- AVFrame based API in swscale
- New bitstream filtering API
  - allows modifying encoded data without decoding it *(e.g. analyzers)*

FFmpeg 5.0 "Lorentz"

Changes - *2*

- Disentangled avformat and avcodec
  - Demuxers no longer embed an entire decoder context
- Frames and packets always reference-counted
- Slice-threaded scaling in swscale
- IMF demuxing
- Dolby Vision metadata
- Hardware-accelerated ProRes and VP9 decoding on MacOS
  - Hardware-accelerated ProRes encoding
- libplacebo
- *Numerous filters and game decoders*

Old API

```
// decoding single video packet in old API
while (pkt->size > 0) {
    //decode pkt into frame, consumes <ret> bytes
    int ret = avcodec_decode_video2(dec, pkt,
frame, &got_output);
    if (got_output)
        <proces output>
    // need to manually offset the buffer
    pkt->data += ret;
    pkt->size -= ret;
}
```

New API

```
// decoding single video packet in new API
avcodec_send_packet(dec, pkt);
ret = 0;
while (ret >= 0) {
    ret = avcodec_receive_frame(dec, frame);
    if (ret < 0)
        break;
    <process frame>
}


 // Subtitles will follow later?
```

FFmpeg 5.1 "Riemann" (summer 2022)

Release

- 130 contributors, ~3k files changed, ~150kLoC touched
- LTS

Changes

- New more flexible and extensible channel layout API
- Expanded fuzzing coverage
- VDPAU-accelerated AV1 decoding
- SIMD optimization for HEVC on ARM64
- JPEG-XL decoding
- Enhanced support for SVT-AV1 encoding options
- Numerous fixes *(and new filters)*

FFmpeg 5.1 "Riemann" (summer 2022)

Channel Layouts API

- Developed since 2013…
  - Did not make it for 5.0
- Arbitrary number of channels in a layout *(Previously only 64)*
- Support for Ambisonics and NGA
- Conceptually a list of channels
- flexible internal representation
  - unspecified ordering
  - mask-based ordering *(same as the old API, WAVEFORMATEX-compatible)*
  - fixed-order ambisonics
  - explicit list of channels *(allows edge cases like dual-mono)*

FFmpeg 6.0 (winter 2023?)

Ongoing

- 191 contributors, ~3.5k files changed, ~220kLoC touched
- Numerous API changes *(and break)*

Current Changes

- FFmpeg CLI multithreading **(WIP),** muxers
- RISC-V optimizations
- AV1 hw decoding for Intel, nVidia & AMD
- New highly-optimized FFT code with SIMD for x86 and ARM
- new API for output of reconstructed frames from encoders *(currently implemented for x264 and libaom)*
- API breaks for deprecations, numerous YUV pix_fmt changes, AVFrame, opacification, channel layouts, H.274

FFmpeg 6.0

Current Hardware Changes

- AV1 hw decoding for Intel, nVidia & AMD
- Hardware-friendly high bit-depth and chroma-resolution pixel formats
- Android MediaCodec though NDKMediaCodec
- Android MediaCodec encoders
- Intel 10/12 4:2:2 & 4:4:4 with VAAPI and QSV + Filters

Codecs

- New Decoders: Bonk, APAC, APAC, Mi-SC4, 100i, VQC, FTR
  PHM, WBMP, XMD ADPCM, WADY DPCM, CBD2 DPCM
- New Filters: adrc, afdelaysr, showcwt, a3dscope
  Ssim360, corr, backgroundkey
- dts2pts bitstream filter: generate timestamps for raw H.264 *(extensible to HEVC and other codecs)*

FFmpeg CLI multithreading

- FFmpeg CLI is used everywhere
  - very flexible
  - architecture still based on the original code from 2000
- Major architectural changes needed
  - code maintainable
  - efficient one-to-many transcoding
  - gathering metrics

- Every component in the transcoding pipeline (demuxing, decoding, filtering, encoding, muxing) will run in its own thread
- Improves latency and new use cases

FFmpeg releases

Concept

- One major per year ABI/API breakage
- One or two minor per year, with no API changes
- Small security releases regularly
- One LTS every 2 year

Plan

• FFmpeg 5.0	January 2022	January 2023
• FFmpeg 5.1 **LTS**	July 2022	July 2024
• FFmpeg 6.0	January 2023	January 2024
• FFmpeg 6.1	July 2023	July 2024
• FFmpeg 7.0	January 2024	
• FFmpeg 7.1 **LTS**	July 2024	

dav1d 1.0 release

Assembly

- ARM32 & ARM64
- x86 32bit
- x86 64bit
    - SSSE3/SSE4
    - AVX2
    - AVX512-IceLake

200kLoC of handwritten ASM!

New Threading

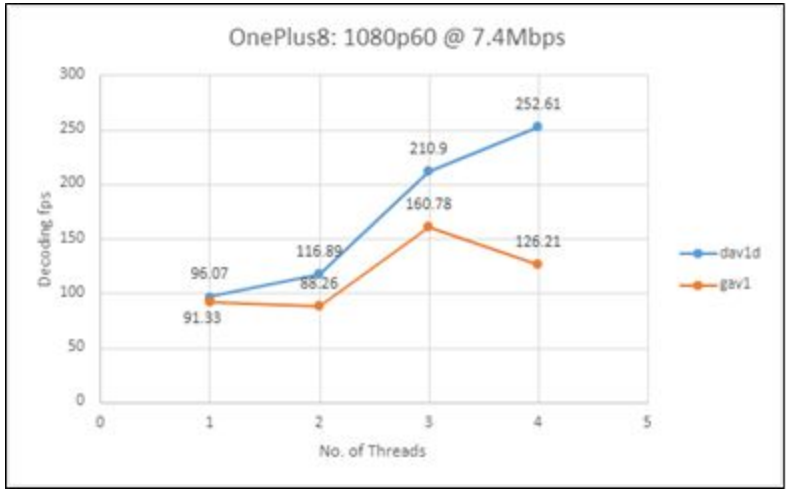
Frame-threads, Slice-threads, Filter threads...

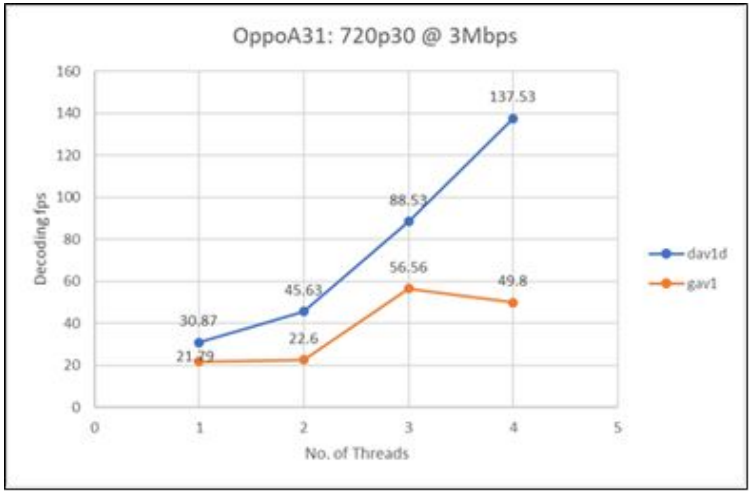
Simpler automatic threading model

Usage

Used by VLC, FFmpeg, Firefox, Chrome, Netflix, Facebook, Apple, Microsoft… Possibly others?

dav1d 1.0 decoding performance





- ST: dav1d is 10-20% (on arm) or ~50% (on x86) faster than gav1
- MT: dav1d's multi-threading scales much better than gav1's, causing it to be **2x** (on arm) or **5x** (on x86) as fast as gav1 with multi-threading on moderate thread counts (4 on arm, 8 on x86) for 1080p content

dav1d 1.1.0 release

1.1.0

- Tons of Fixes, and compliance to samples (Argon), and race conditions
- Function to query decoder frame delay
- AVX2 12bpc asm
- AVX-512 cdef, loopfilters, itx + bugs
- NEON z1/z3 8bpc
- Fix overflows and clipping in a large number of ASM code + fuzzing

VLC.js

Demo

VLC.~~js~~wasm

Demo

Thanks!

Any questions?

jb@videolan.org