AV1: One Year Later

Nathan Egge <negge@mozilla.com>
Brion Vibber <bvibber@wikimedia.org>
Mile High Video - July 30, 2019
Slides: https://xiph.org/~negge/MHV2019.pdf

AV1 Ecosystem Update at NAB 2019

The Rapid Progress Of The AV1 Video Format Over The Past Year

Written by Michael Larabel in Mozilla on 16 April 2019 at 12:08 AM EDT. 8 Comments

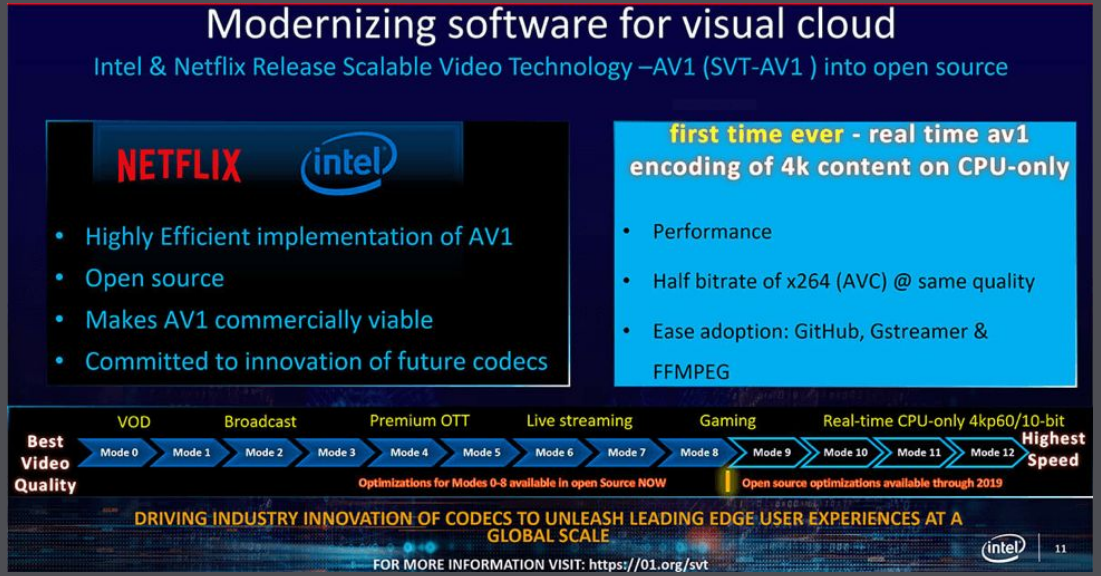
Year in review NAB 2018 to NAB 2019

- Hard to cover in just 30 minutes

[1] https://www.nabstreamingsummit.com/session/av1/

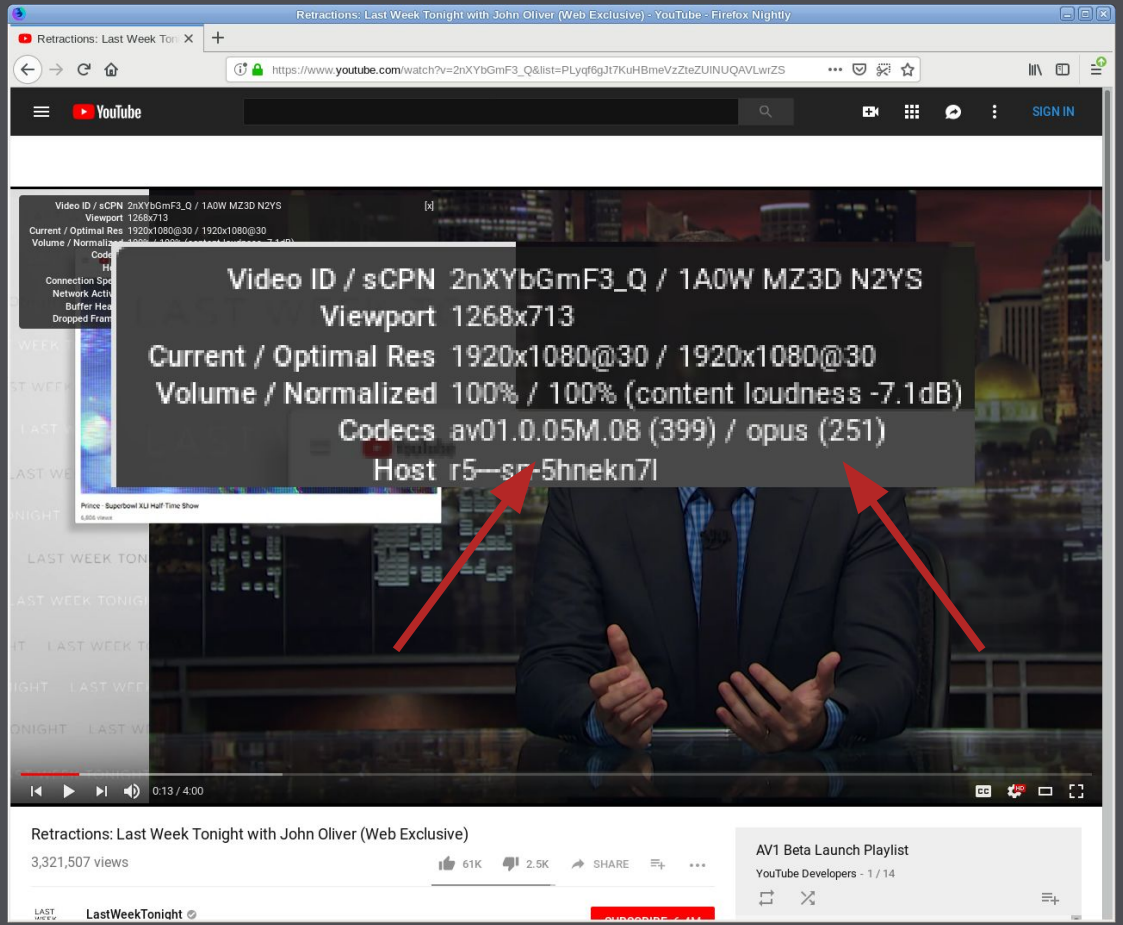
[2] https://www.phoronix.com/scan.php?page=news_item&px=AV1-Progress-Over-2019

SVT-AV1 Announced at NAB 2019



- Open Source, BSD license
- 3 different mechanisms for parallelism
- 13 different speed settings (planned)

AV1 + Opus Content on YouTube at IBC 2018



Firefox 65

- Non-Windows behind flag:
  `media.av1.enabled = true`

Chrome 70

- Enabled by default on desktop platforms and Android

Edge 18

- Enabled with AV1 Video Extension from Microsoft Store

AV1 software decoders on billions of devices in October 2018!!!

AV1 (libaom) decoding is slow (July 7, 2018)

moz://a

AV1 vs VP9 vs AVC (h.264) vs HEVC (h.265): Part IV - Decode

- July 07, 2018

Codec	Decode			
	real time	vs x264	CPU time	vs x264
x264	0.409s	1:1	1.059s	1:1
x265	0.980s	2.40:1	1.813s	1.71:1
VP9	0.761s	1.86:1	1.253s	1.18:1
AV1	4.178s	10.22:1	3.993s	3.77:1

Codec	Decode			
	real time	vs x264	CPU time	vs x264
x264	0.631s	1:1	1.856s	1:1
x265	1.190s	1.89:1	3.024s	1.63:1
VP9	0.780s	1.24:1	2.095s	1.13:1
AV1	11.750s	18.62:1	11.703s	6.31:1

"At of [sic] June 2018, the AV1 decoder can only decode in a **single thread**. Decoders for H.264, H265, and VP9 utilize multi-threads."

"To decode the files encoded with AV1, the CPU time was approximately **3.5-6x longer** than that to decode files encoded with x264. This means that it required **3.5-6x more CPU** power to decode files encoded with AV1."

[1] https://www.texpion.com/2018/07/av1-vs-vp9-vs-avc-h264-vs-hevc-h265-4-decode.html

Optimized Decoding

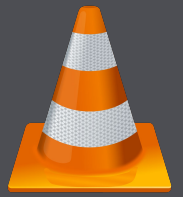
- AV1 reference (libaom) is not designed for decoder performance
  - Large binary, too much memory, missing SIMD, research codebase

- AOMedia asked for proposals for an open-source optimized decoder

```
Requirements:
+ Open Source (MIT/BSD licensed, or similar)
+ Full SIMD for x86-32, x86-64, ARMv7, and ARMv8.
+ Multi-threaded
+ Bit-identical decoding of all features vs. libaom
```

- French non-profit VideoLAN selected

- Dav1d is an AV1 Decoder (dav1d) project is born



dav1d Project

VideoLAN / FFmpeg / Two Orioles

Goals

- Small binary size
- Low CPU utilization
- Multi-threaded
- Cross-platform

Implementation

- C99 (no VLA, complex, GNU extensions)
- ASM (no intrinsics, ASM files like FFmpeg)

License

- BSD (like opus, libvorbis, etc.)

Project Timeline:

2018-Jun-29: AOM decoder CfP
2018-Jul-13: Deadline to submit
2018-Jul-13: VideoLAN selected
2018-Aug-22: dav1d contract executed
2018-Sep-22: dav1d open sourced @ VDD 2018
2018-Nov-11: dav1d in Firefox Nightly
2018-Dec-11: dav1d-0.1.0 Release
2018-Dec-27: dav1d in vlc-3.0.5
2019-Mar-13: dav1d-0.2.1 Release
2019-Mar-29: dav1d in Chrome Canary
2019-Apr-24: dav1d-0.1.0 in Chrome 74
2019-May-21: dav1d-0.2.1 in Firefox 67

dav1d Project

VideoLAN / FFmpeg / Two Orioles

Goals
- Small binary size
- Low CPU utilization
- Multi-threaded
- Cross-platform

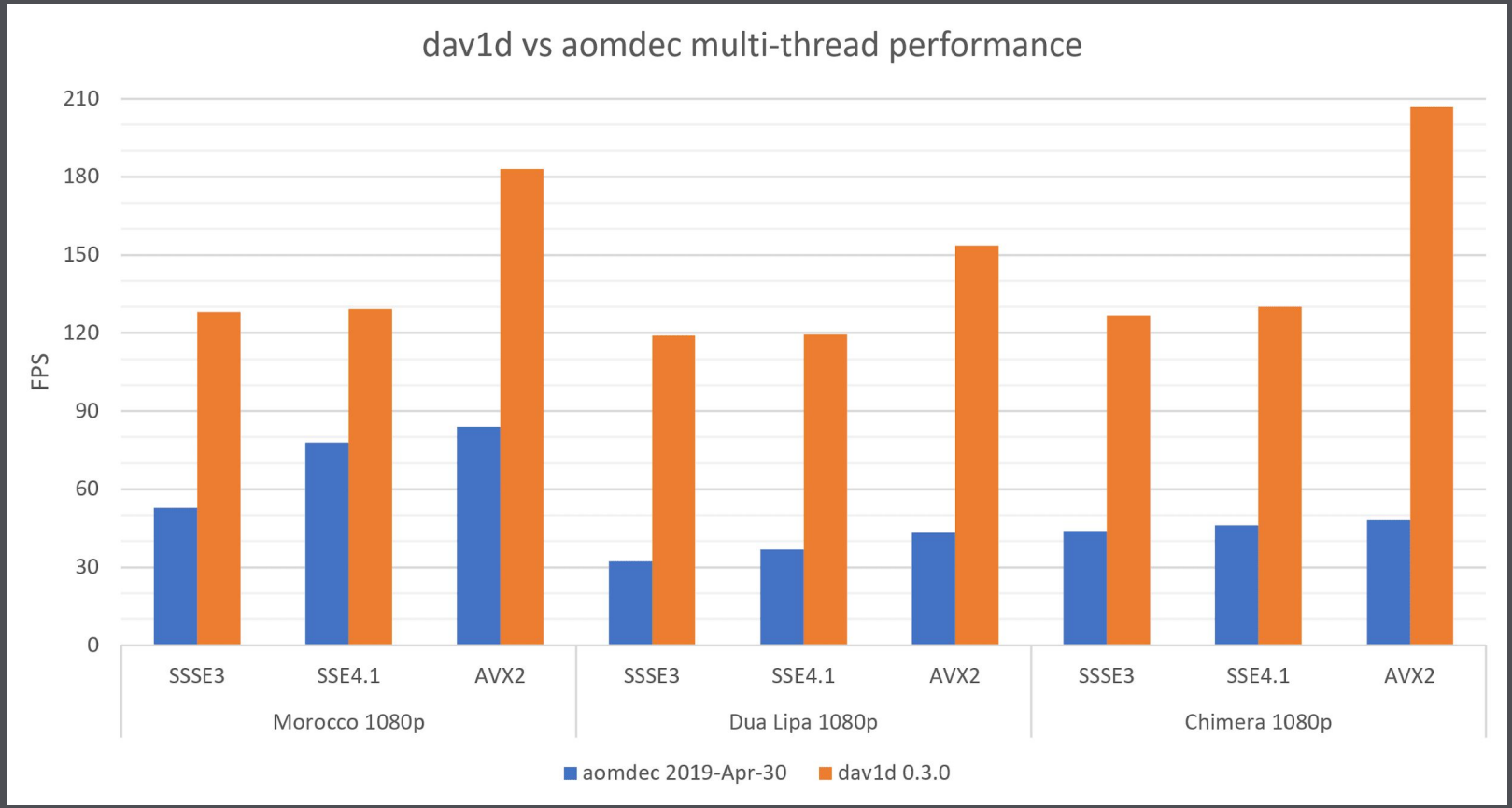
Implementation
- C99 (no VLA, complex, GNU extensions)
- ASM (no intrinsics, ASM files like FFmpeg)

License
- BSD (like opus, libvorbis, etc.)

Project Timeline:

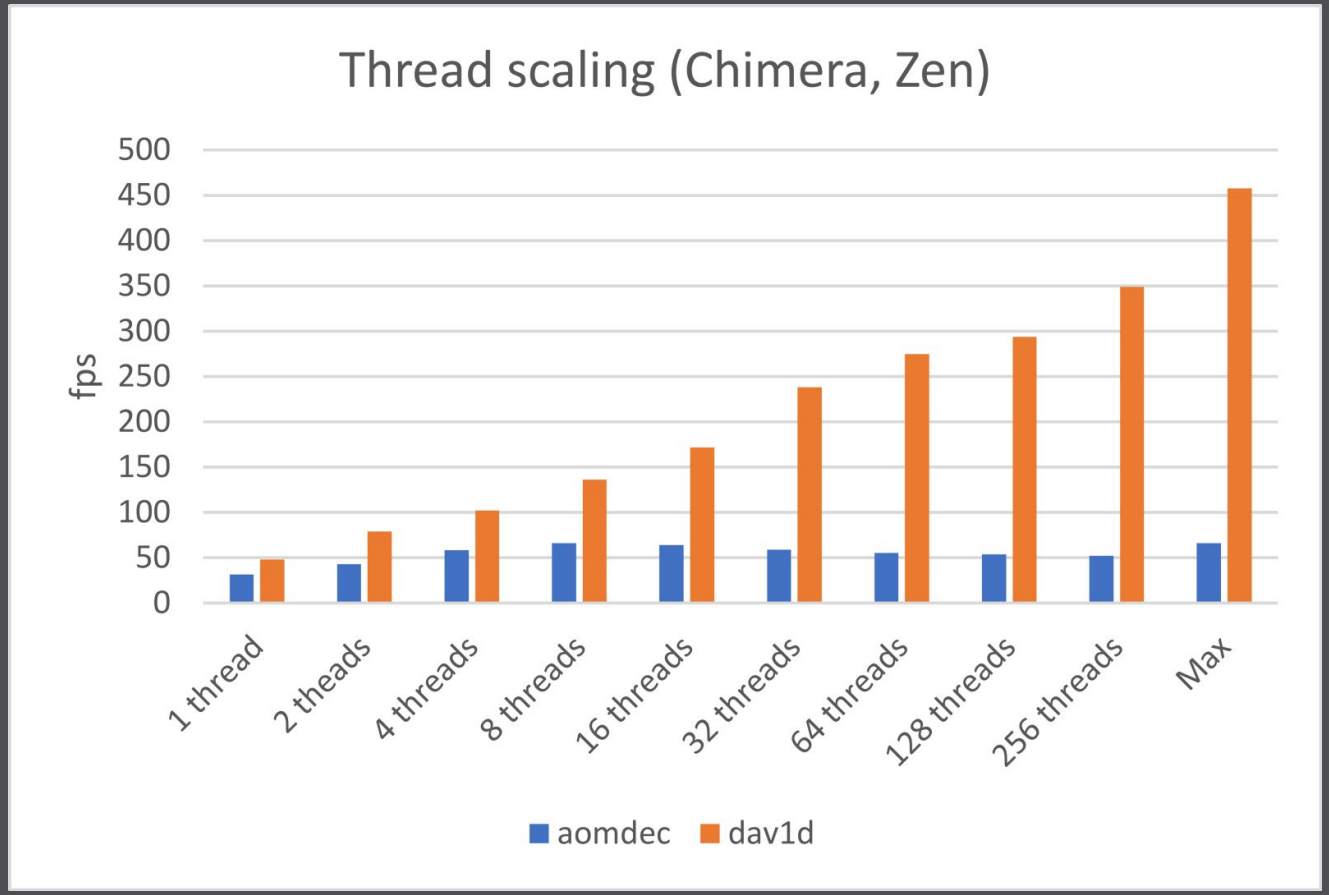
2018-Jun-29: AOM decoder CfP
2018-Jul-13: Deadline to submit
2018-Jul-13: VideoLAN selected
2018-Aug-22: dav1d contract executed
2018-Sep-22: dav1d open sourced @ VDD 2018
2018-Nov-11: dav1d in Firefox Nightly
2018-Dec-11: dav1d-0.1.0 Release
2018-Dec-27: dav1d in vlc-3.0.5
2019-Mar-13: dav1d-0.2.1 Release
2019-Mar-29: dav1d in Chrome Canary
2019-Apr-24: dav1d-0.1.0 in Chrome 74
2019-May-21: dav1d-0.2.1 in Firefox 67

dav1d-0.3.0 Release (May 3, 2019)

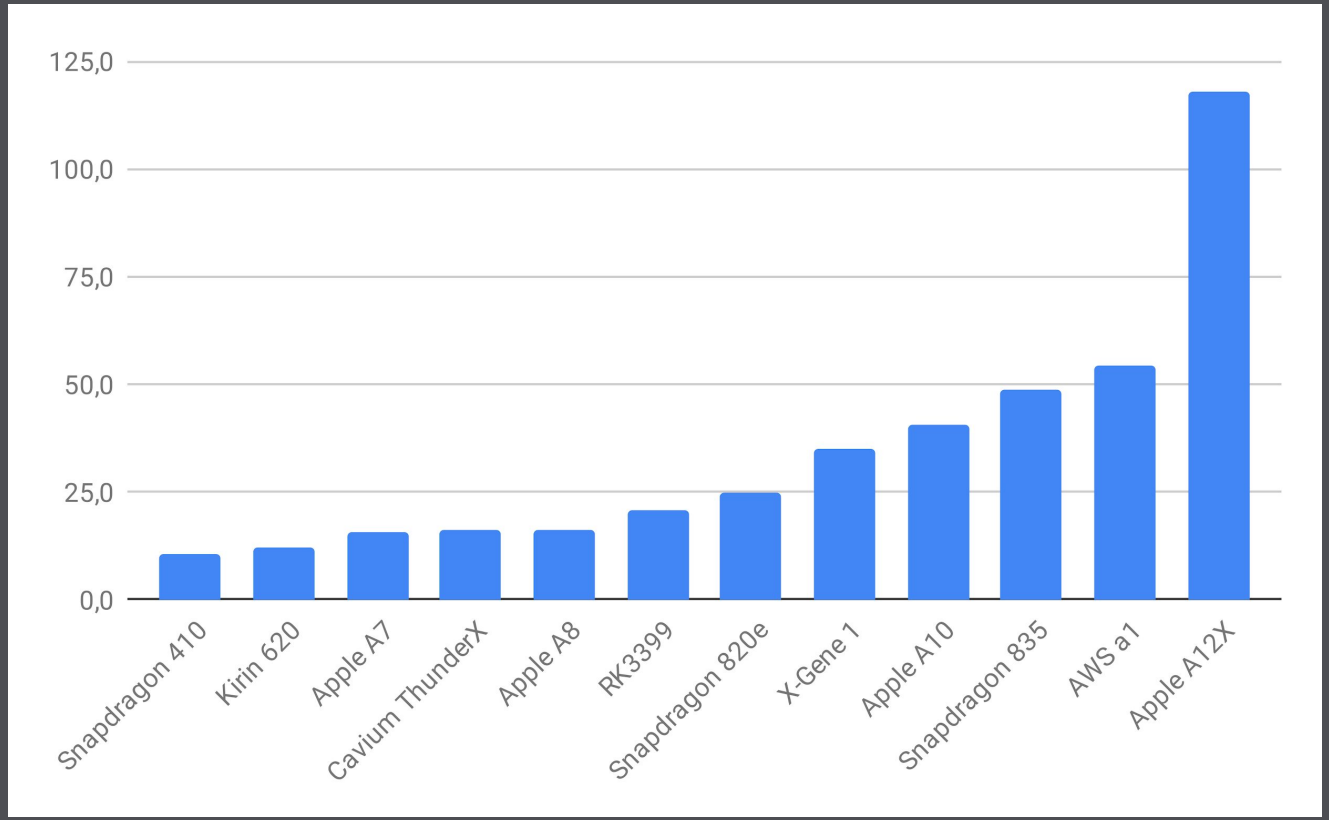


[1] https://hacks.mozilla.org/2019/05/firefox-brings-you-smooth-video-playback-with-the-worlds-fastest-av1-decoder

dav1d is Scalable!



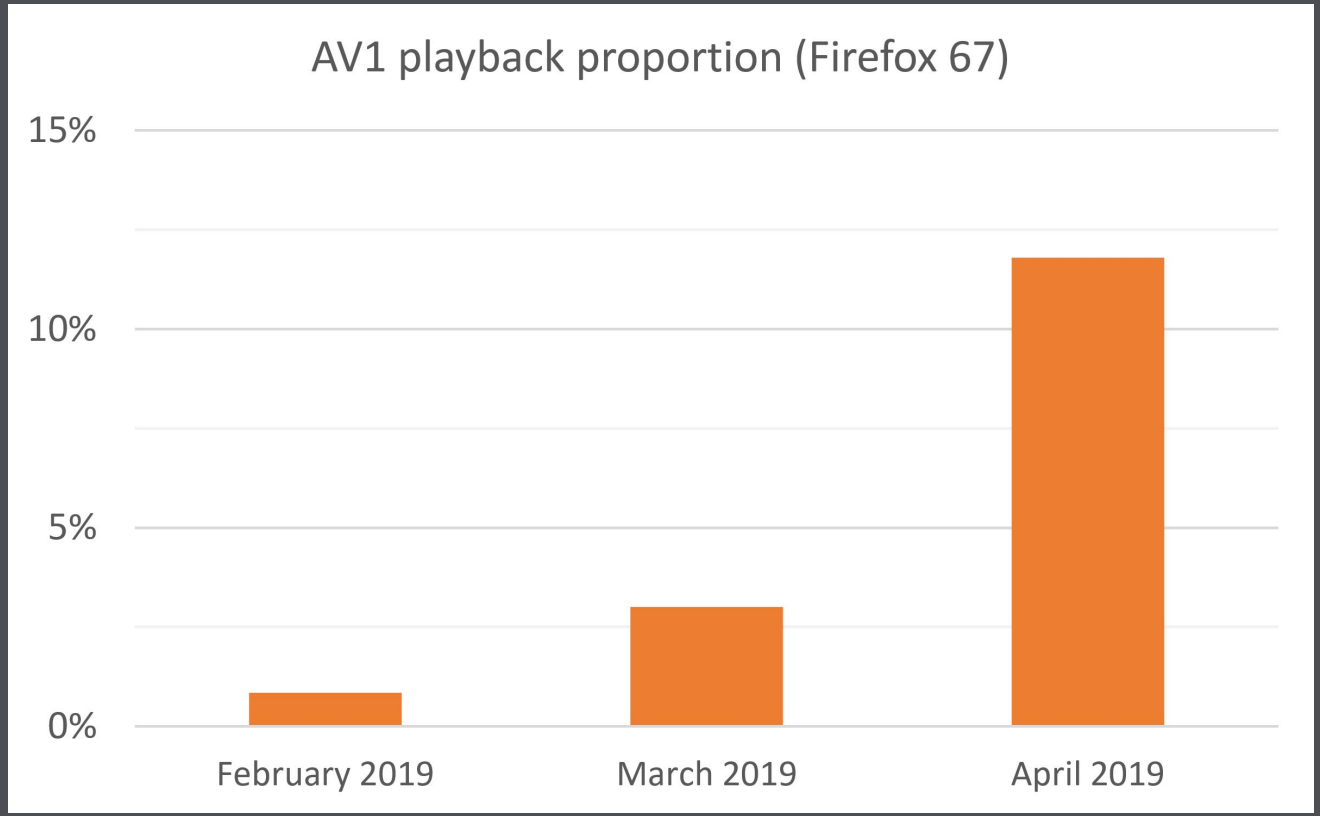
dav1d Can Decode 4k on Mobile(*)



1080p decode speeds in FPS

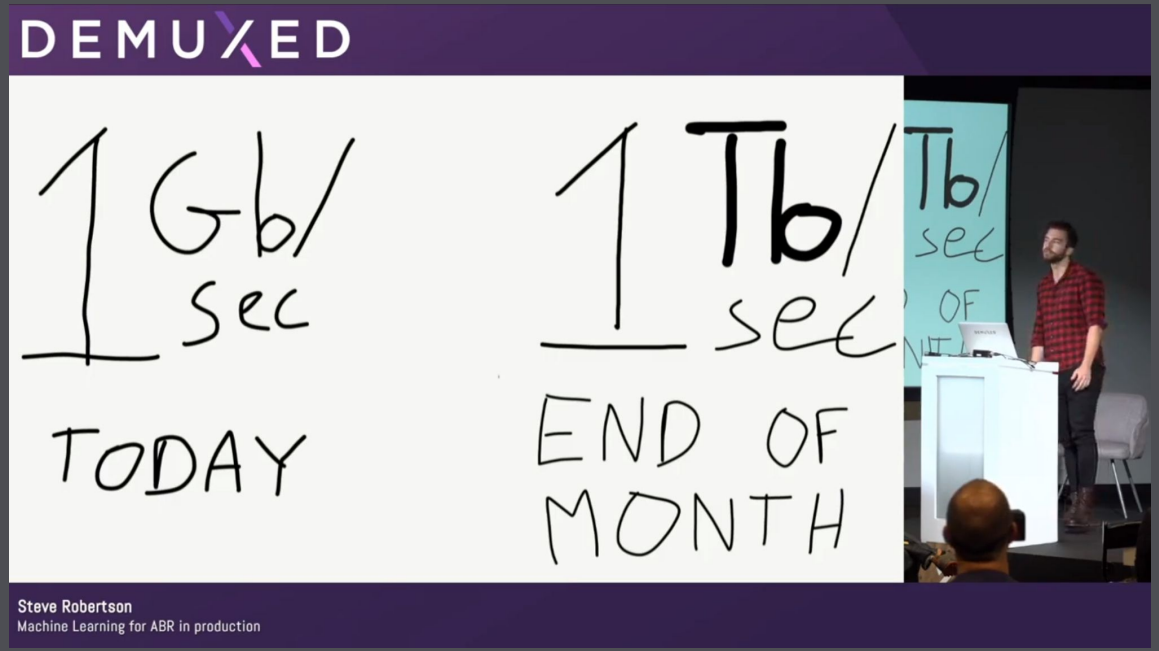
(*) If you are Apple

AV1 already seeing adoption on the web!!



[1] https://hacks.mozilla.org/2019/05/firefox-brings-you-smooth-video-playback-with-the-worlds-fastest-av1-decoder

AV1 Update - Demuxed (October 18, 2018)

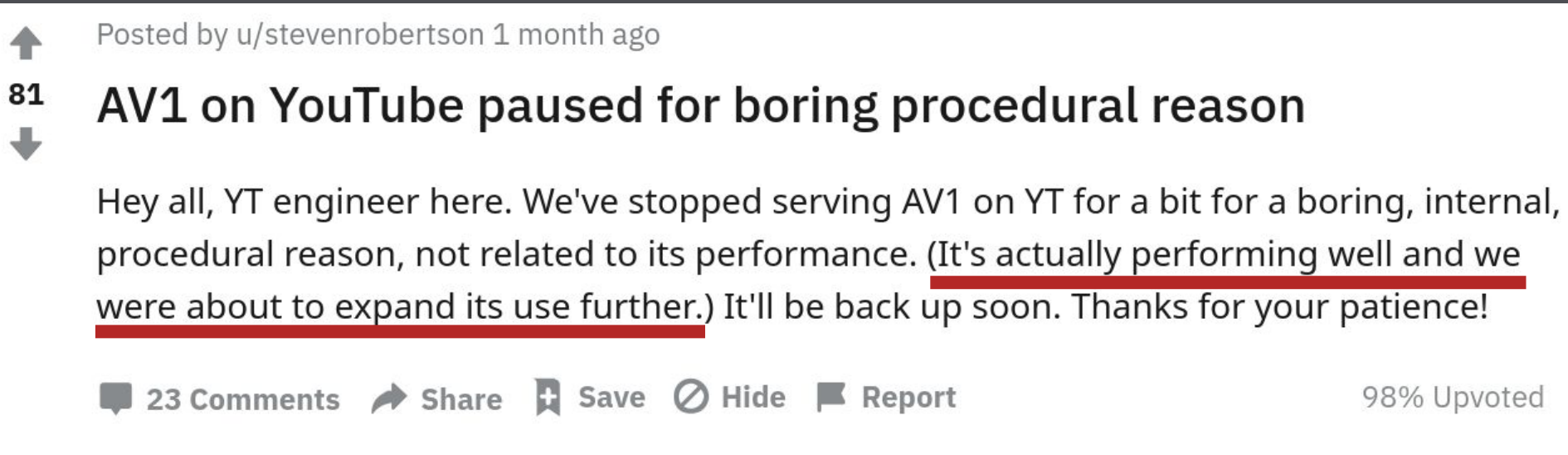


Steve Robertson (Google / YouTube)

- "Project will be **streaming 1 Tb / sec** AV1 by the end of the month"
- "Small fraction of YouTube's egress, sign that **it is working for us in production**"
- "**Not a cost effective choice** at this time"
- "We are doing this to make sure people know we are **deadly serious about AV1** and are committed to its success"

[1] https://www.youtube.com/watch?v=iuaWhmETcRE&t=65

YouTube Pauses AV1 Deployment



Posted by u/stevenrobertson 1 month ago

81

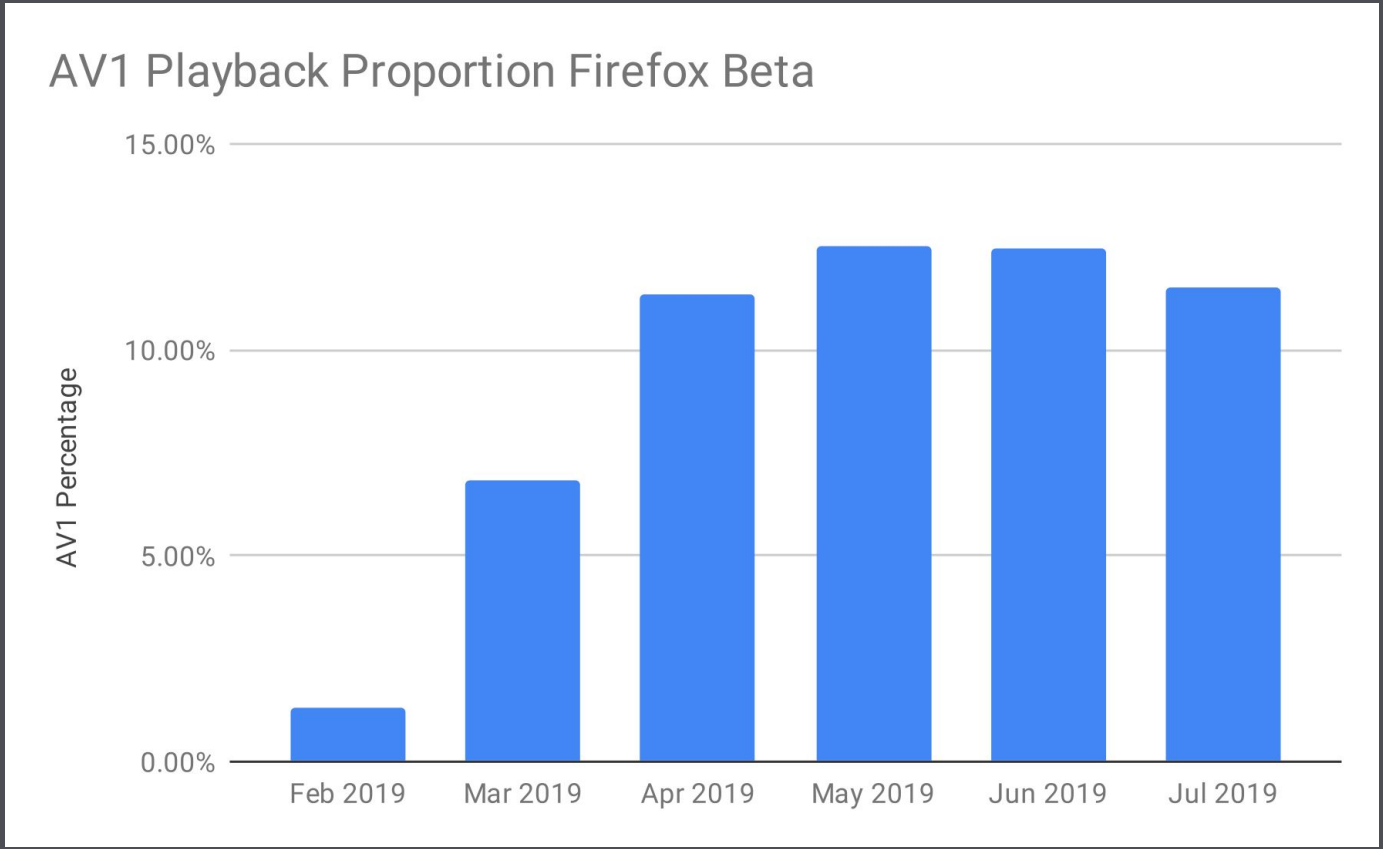
AV1 on YouTube paused for boring procedural reason

Hey all, YT engineer here. We've stopped serving AV1 on YT for a bit for a boring, internal, procedural reason, not related to its performance. (It's actually performing well and we were about to expand its use further.) It'll be back up soon. Thanks for your patience!

23 Comments Share Save Hide Report 98% Upvoted

[1] https://www.reddit.com/r/AV1/comments/c0c7e9/av1_on_youtube_paused_for_boring_procedural_reason/

Nearly 1 in 8 Videos Played are AV1 in July 2019!!

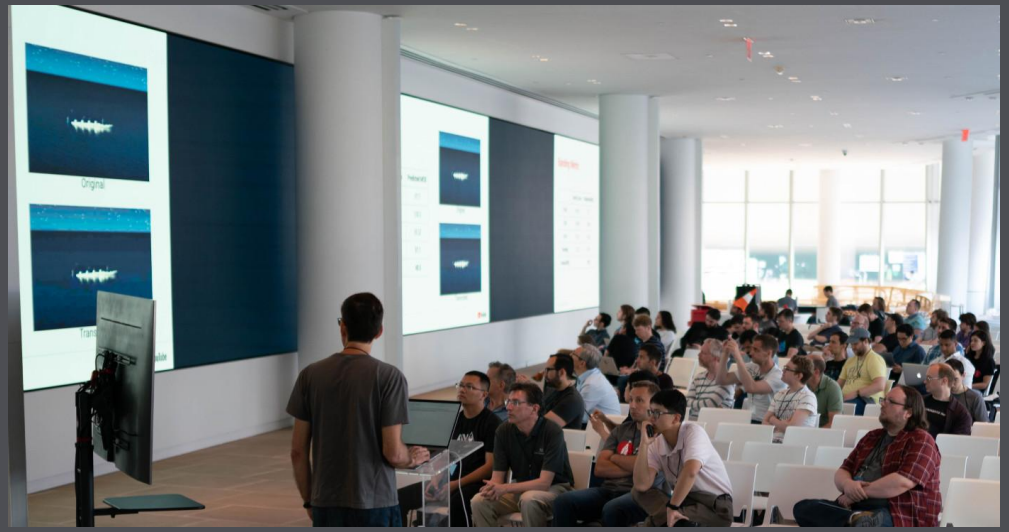


Big Apple Video 2019 Conference

Mozilla and Vimeo co-hosted event

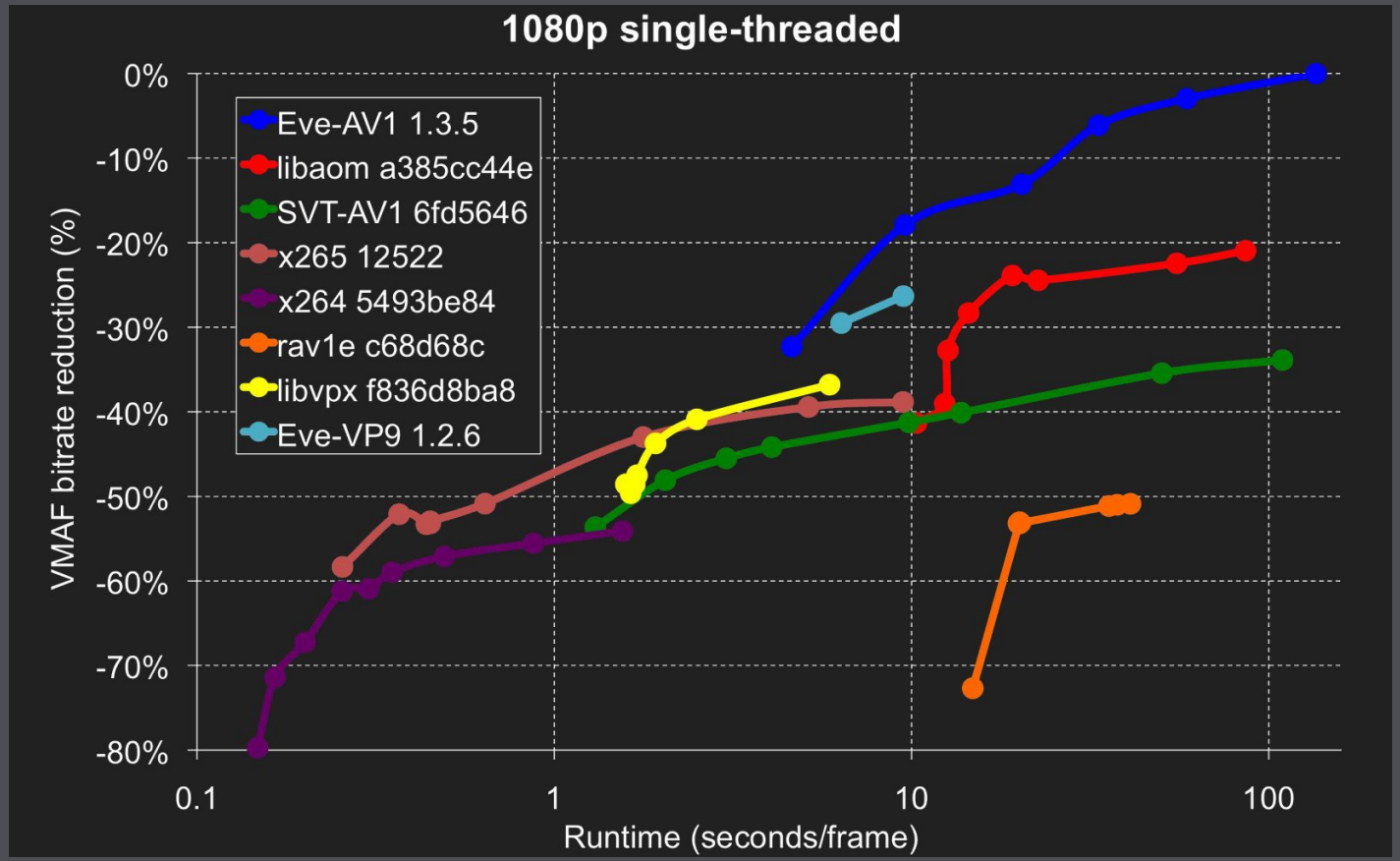
- Full day of talks, streamed live by Vimeo
  - 120 in person and 1000 remote participants

Big Apple Video 2019



[1] https://bigapple.video
[2] https://twitter.com/bigapplevideo
[3] https://vimeo.com/bigapplevideony

Two Orioles @ Big Apple Video 2019



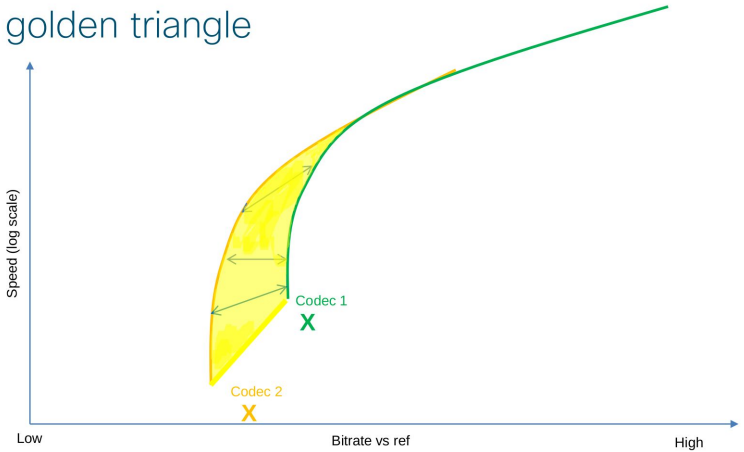
[1] https://bigapple.video/slides/TwoOrioles-BAV2019.pdf

Cisco @ Big Apple Video 2019

Why AV1? A new codec world

- H.264 is 16 years old we need a step-change in capability
  - AV1 provides that in a way we can use
- Supports mixed and next-gen content
- All tools in one profile
  - Video, screen content, AR/VR, scalability ..

The golden triangle



"In our demo, we not only managed to encode live 720p30 camera video at **half the bandwidth of H.264**, but we also encoded high frame rate share at 1080p30 using around **2/3 of the bitrate of H.264 encoding 720p30**, all encoded on a commodity laptop."

[1] https://bigapple.video/slides/Cisco-BAV2019.pdf
[2] https://blogs.cisco.com/collaboration/cisco-leap-frogs-h-264-video-collaboration-with-real-time-av1-codec

AV1 Encoder Ecosystem

- Open Source Encoders:
  - Google - libaom [1]
  - Intel & Netflix - SVT-AV1 [2]
  - Mozilla & Xiph - rav1e [3]

- Closed Source Encoders:
  - Two Orioles - EVE for AV1
  - Socionext (FPGA offload)
  - NGCodec (FPGA intra only)
  - ATEME
  - Harmonic
  - Bitmovin (cloud encoder)
  - Visionular - Aurora
  - Allegro DVT (hardware)
  - Cisco AV1
  - Every hardware vendor in AOM
  - Probably more I don't know of

[1] https://aomedia.googlesource.com/aom/
[2] https://github.com/OpenVisualCloud/SVT-AV1
[3] https://github.com/xiph/rav1e

AV1 is Everywhere

- ISO-BMFF [1]
- TS [2]
- MP4 / MKV
- HEIF (aka AVIF) [3]
- RTP [4]
- FFmpeg [5]

[1] https://aomediacodec.github.io/av1-isobmff/

[2] https://aomediacodec.github.io/av1-isobmff/

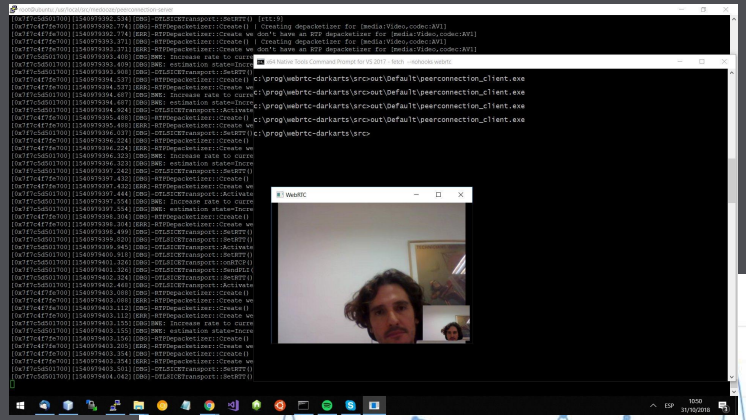
[3] https://aomediacodec.github.io/av1-avif/

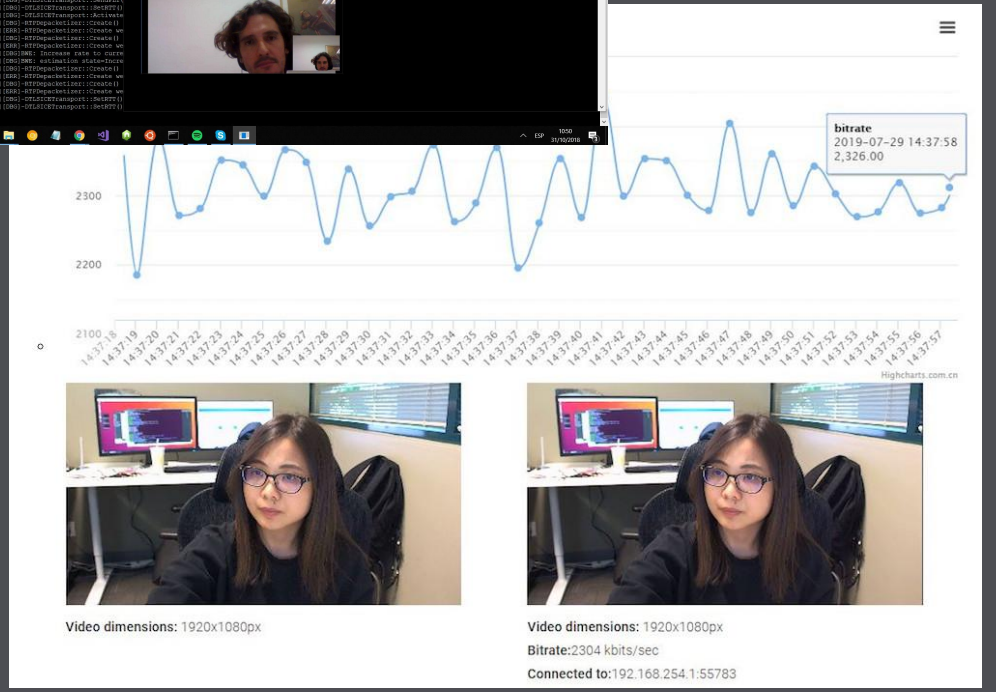
[4] https://github.com/AOMediaCodec/av1-rtp-spec

[5] https://trac.ffmpeg.org/wiki/Encode/AV1

AV1 in WebRTC

- Cosmo demo at IETF 105
  - Using RT libaom
  - Test binaries available
- Visionular blog post
  - 1080p support
  - Test binaries available





[1] https://visionular.com/aurora_rtc

[2] https://twitter.com/murillo/status/1057603090910797825

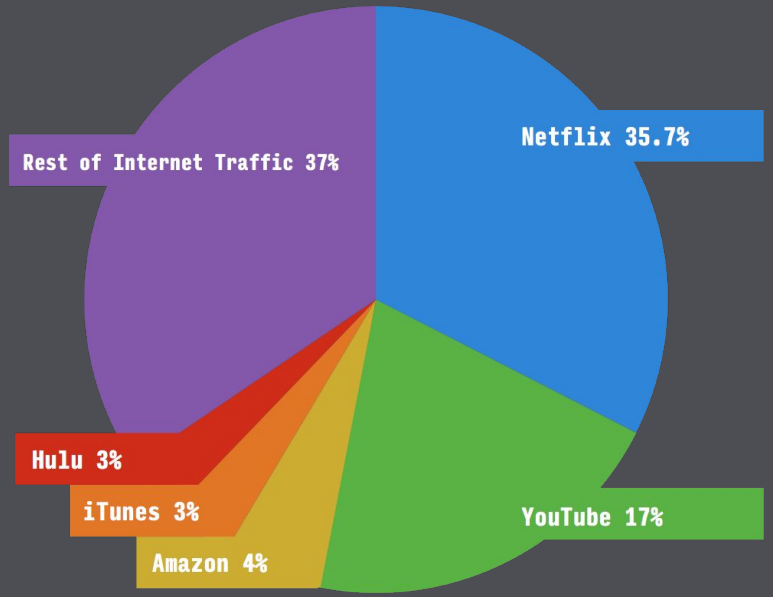
[3] http://webrtcbydralex.com/index.php/2019/07/09/real-time-av1-in-webrtc-is-now-production-ready/

Internet Video Traffic

YouTube
- 1 billion hours (114,155 years) watched per day
- Great Pyramid of Giza: ???
- 60% of it on mobile

Facebook
- Millions of videos uploaded every day
- Billions of videos watched every day

And videos are getting bigger
- 4k
- HDR



82% of Internet traffic by 2021 [1]

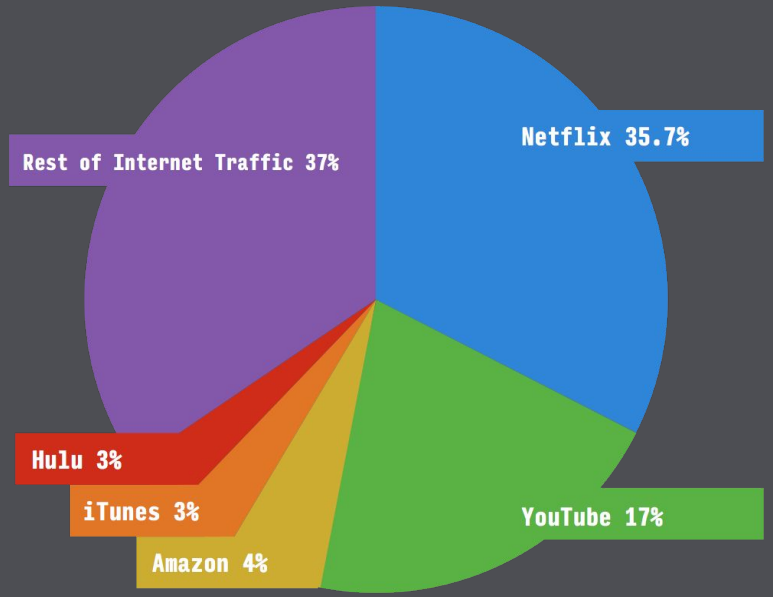
[1] https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html

Internet Video Traffic

YouTube
- 1 billion hours (114,155 years) watched per day
- Great Pyramid of Giza: **131,200 man-years!** [2]
- 60% of it on mobile

Facebook
- Millions of videos uploaded every day
- Billions of videos watched every day

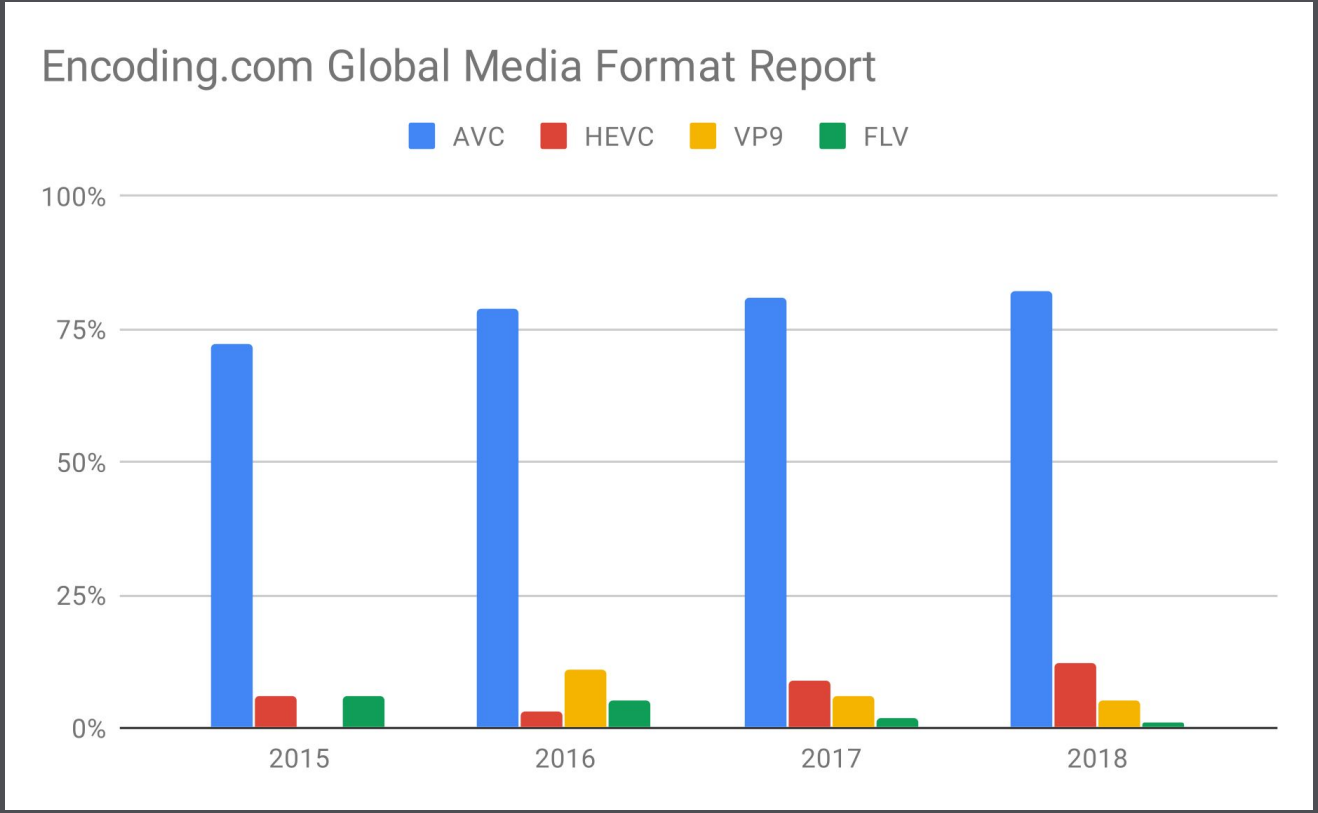
And videos are getting bigger
- 4k
- HDR



82% of Internet traffic by 2021 [1]

[1] https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html
[2] https://web.archive.org/web/20070608101037/http://www.pubs.asce.org/ceonline/0699feat.html

Consider Just the Broadcast Space



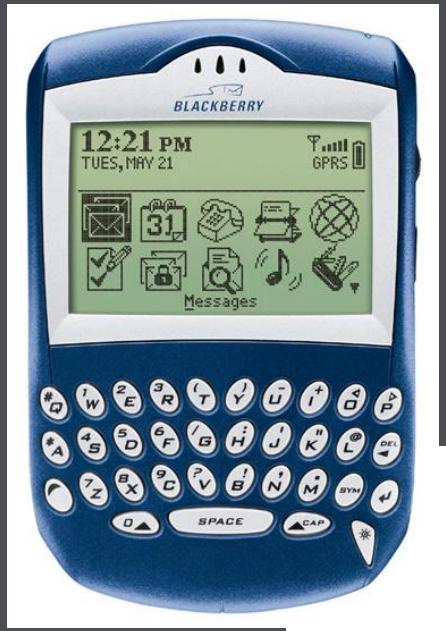
	AVC (%)	Ingest (PB)
2014	69	0.95
2015	71	1.45
2016	79	5.9
2017	81	12.3
2018	82	19.3

[1] https://www.encoding.com/resources/

Surprising result?

- H.264 was standardized in 2003

- Other things that were new in 2003
  - Blackberry Quark 6210
  - Thinkpad X31



Why is so much content encoded in H.264?

- It works everywhere!
- Open source ecosystem
  - x264 is the best H.264 encoder
- Mature workflows for content creation
- Momentum in the industry
- Predictable licensing model

What's wrong with just using H.264?

- Bitrate too high
  - Delivery costs can exceed encoding on head content
  - Unable to reach developing parts of the world
- Lack of new experiences
  - 4k, HDR, 360
  - VR/AR have even higher quality requirements
  - Cloud gaming?
- Already reaching the capacity of the internet
  - What happens when demand outpaces bandwidth?

Why haven’t VP9 or HEVC addressed this need?

- One format does not play everywhere
- Poor open source encoder ecosystem
  - Lack of VP9 ecosystem hurt adoption
  - x265 / SVT-HEVC not as mature x264
- Long product cycle in broadcast industry
- Unpredictable license model

- VP9 and HEVC are better technically, but each had non-technical failings

Demo

Demo: Live-streaming AV1 with open source tools

Live encoding / streaming demo with multiple clients.

Toolset:

- **ffmpeg**

Encoder:

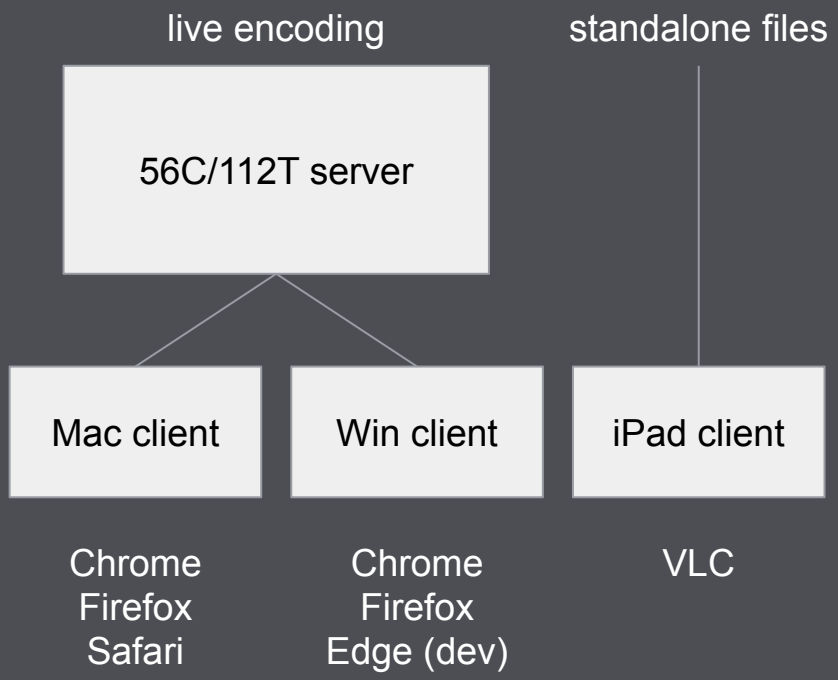
- **SVT-AV1** (needs patch to ffmpeg)

Muxer:

- ffmpeg's **DASH** muxer (patched for AV1)

Client tools:

- **dash.js**
- **ogv.js** for Safari
- **VLC** for iOS

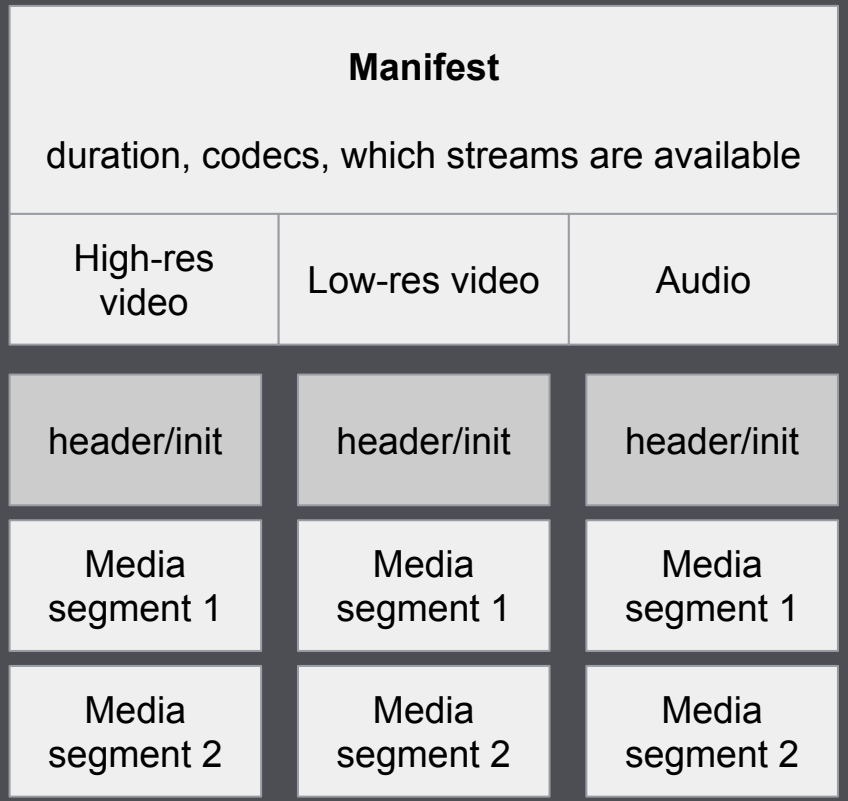


Demo: quick intro to DASH streaming

Adaptive streaming systems such as DASH and HLS work by splitting a complex set of data streams into manageable chunks, with manifest files referencing them.

The client pulls down the manifest, then fetches chunks as needed -- if necessary updating the manifest itself when doing live encodings.

In VOD the chunks may be stored conveniently in a single file, but for live encoding they usually live separately.



Demo: SVT-AV1

You can use libaom for VOD but other encoders are faster, needed for live encoding.

Open-source SVT-AV1 encoder is scalable across tiles, frames, and wavefront management of macroblocks.

While still under active development, it's able to provide live 1080p60 encoding on a powerful enough machine at data rates that compare very well against today's king, h.264 -- or multiple resolutions at lower frame rates.

Notes:

- scales to many cores (eg, **112 threads**)
- use **fastest encoding mode** for live encoding, or adjust the mode for better VOD bitrates
- an **ffmpeg patch** is provided, but needs a lot of work
- we **modified** the ffmpeg patch to provide the encoding options and packetization we needed

We hope to **merge changes upstream** for other users.

Demo: ffmpeg DASH muxer

ffmpeg's DASH muxer can produce MPEG-DASH or HLS-compatible output. We had to make one small modification to support AV1 codec parameters in the manifest file.

Notes:

- produces DASH manifest and chunk files, using **fragmented MP4** by default
- can produce HLS additionally (or the HLS muxer works in a pinch)
- the produced files can be sent off to a server or served directly off the filesystem
- codec string bug for AV1: https://trac.ffmpeg.org/ticket/8049 (temporary patch used)

Once the codec string bug is fixed, producing DASH manifests with AV1 video should work reliably.

Demo: dash.js

DASH streaming in browsers is mediated by JavaScript code that interfaces with the Media Source Extensions API. There are several packages which provide high-level DASH streaming; for this demo we used **dash.js**.

This works for AV1 in Chrome, Firefox, and Edge (dev) with no modifications.

Demo: ogv.js and Safari

Safari, the default browser on macOS does not currently support AV1 natively. At Wikipedia we use a **WebAssembly** shim called **ogv.js** for playback of non-native video formats.

ogv.js includes a build of the **dav1d** AV1 decoder, which performs well enough in Safari at low resolutions but is limited by a lack of threading and SIMD processing -- both things that are being improved in the web platform, with experimental support for both available in Chrome.

As WebAssembly threading and SIMD support matures, this will become a more practical option for browsers without native AV1 support.

Demo: VLC and iOS

For iOS devices, Safari does not support AV1 but playback in third-party apps is possible. Like Chrome and Firefox, VLC uses the **dav1d** decoder, which is getting increasingly optimized on ARM64 as well as x86.

Notes:

- VLC is available in embedded form for iOS apps as “**MobileVLCKit**”, which may provide a convenient tool for application developers to integrate performant AV1 playback.

moz://a

End Demo

AV1 Has The Potential To...

- Use one next-generation format on all devices
- Have good open-source production implementations
- Take advantage of industry momentum
- Replace aging encoder infrastructure
- Predictable licensing model

moz://a

Questions?

Considering that video and images make up about 80% of all internet traffic, the impact of how things get encoded is pretty big. Even a modest 1% BDR gain tool translates into about 20 EB of traffic yearly currently, or 20,000,000,000 GB.

Steinar Midtskogen (Cisco)