Encoding at Netlfix

May 2014

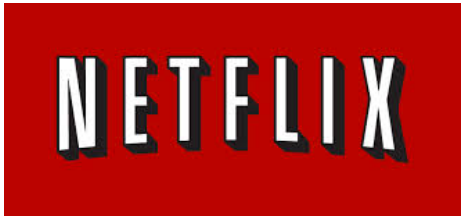
David Ronca, Manager Encoding Technology

Introduction

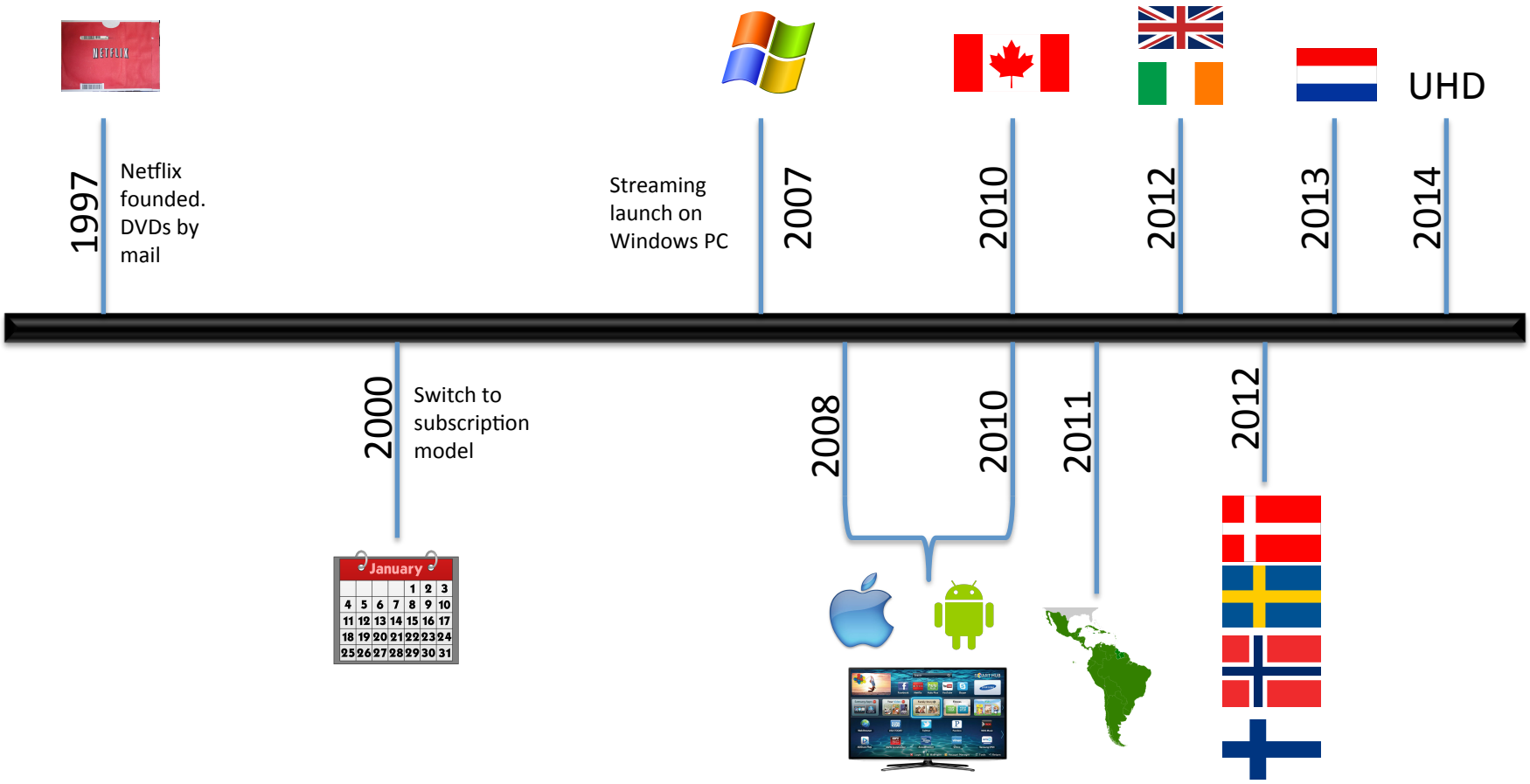
Netflix started streaming in 2007 with IE-plugin hosting Windows Media Player and ~2000 titles

Today …

- Award-winning Netflix Originals
- Tens of millions of active Netflix devices
- Billions of hours of content streamed
- International
- Recently launched 4K UHD service



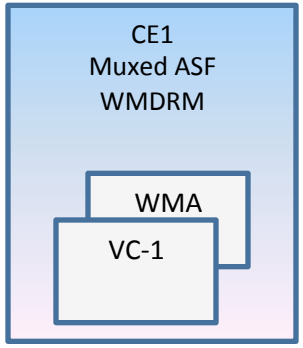
Netflix Timeline

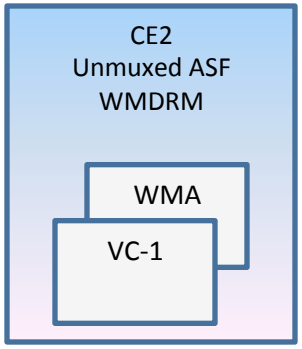


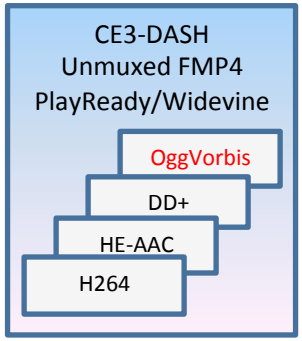
Discussion Topics

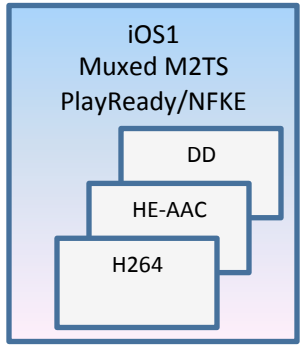
- Brief Historical look at our encoding workflows
- Robust and scalable encoding in the cloud
- Our 4K Solution
- Netflix Source Format Roadmap

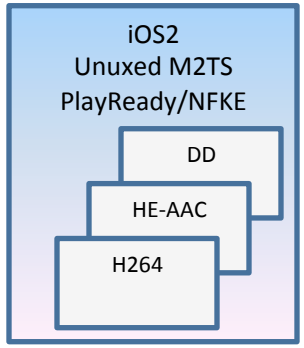
Streaming Profile Review

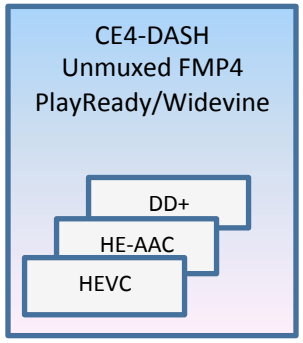




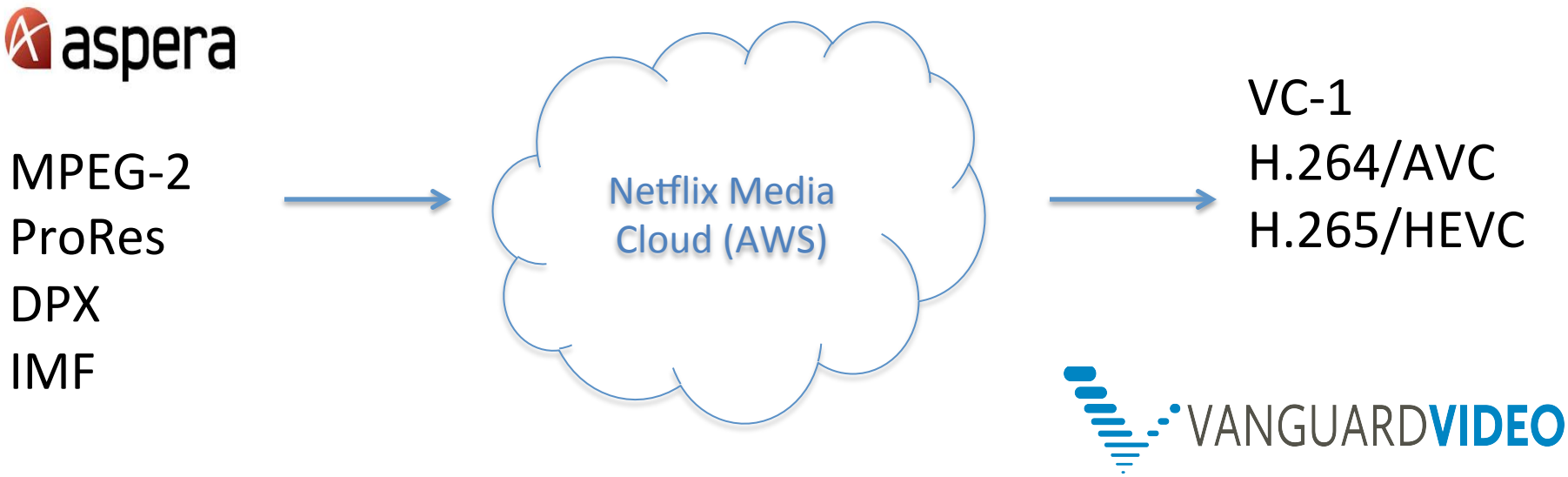








Netflix Media Processing At A Glance



- Distributed architecture
- Elastic
- 100% Cloud
- Thousands of EC2 instances

Netflix Encoding V1 (2007)

- Script files running on servers in data center
- 100% MSFT: WMV, WMDRM
- Encoding framework built on AviSynth and VFW
- Scale to a few thousand titles

Netflix Encoding V2 (Matrix, 2009)

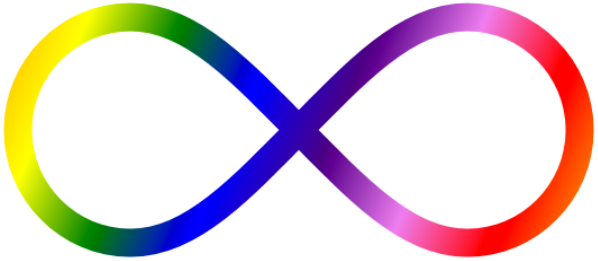
- Move to EC2 (elastic)
- Written in Java
- Same video framework (AviSynth and VFW)
- Scale to tens of thousands of titles
- Added H264, AAC, DD+, etc.
- Production for all encode profiles starting with “Vega” BD-Live

Matrix Encoding Limitations

- Very long encode times (10-40x play time)
- Inadequate source inspections
  - Loose tolerances
  - Unreliable duration
  - Few source problems detected early
  - Problem source == problem encodes
- Source problems delivered to customers
- Inability to scale sufficiently for growing business

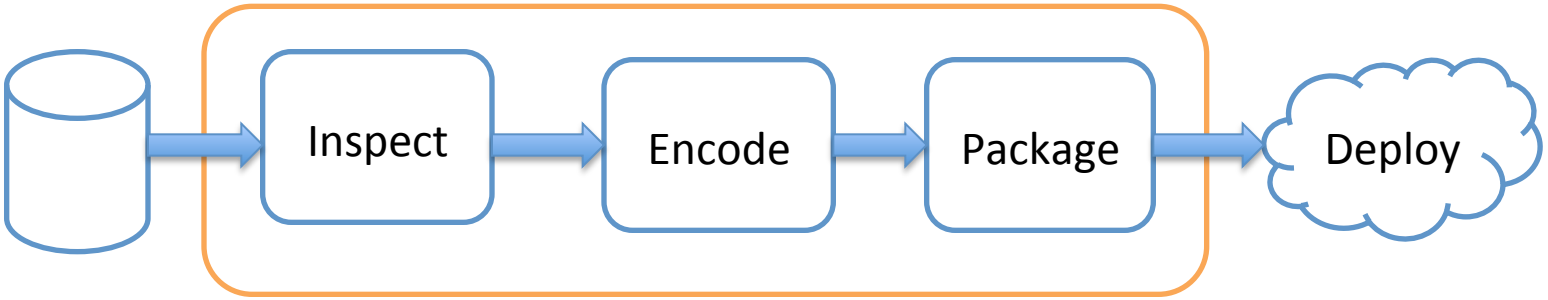
Netflix Encoding V3 (Reloaded, 2013)

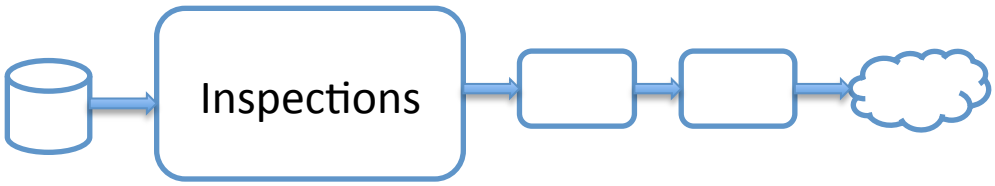
- Linux & ffmpeg
- Designed for infinite scaling
- Robust source inspections
- Parallel video encoding
- Very large source files (i.e. 4K)



Designing for Scale

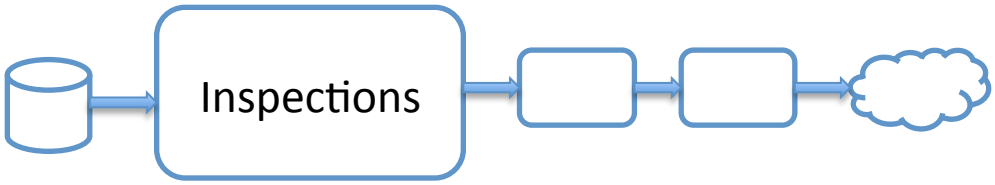
- Thorough inspection of all incoming assets
- Issues detected at point of failure
- Robust and failure tolerant
- Progressive model for long processes
- Parallelization where needed
- Highly predictable output





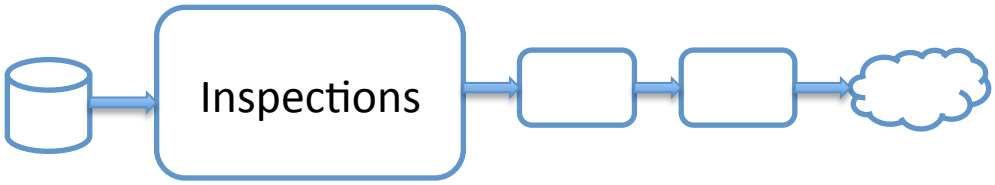
A predictable encoding system requires good input

- Verify source conformance to delivery specification
- Auto-route failures with actionable error messages
- System layer and decode inspections
- Fully index elementary streams
- Multiple sources of truth



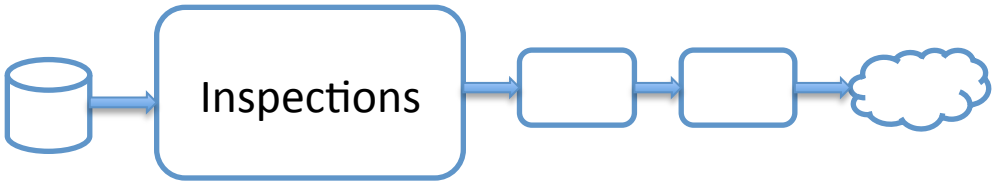
System layer inspections

- Verify conformance to relevant system specification (M2TS, MP4, etc.)
- Example: M2TS
  - Packet inspections: synch, continuity
  - Timing: PCR, buffer, STD delay, etc.



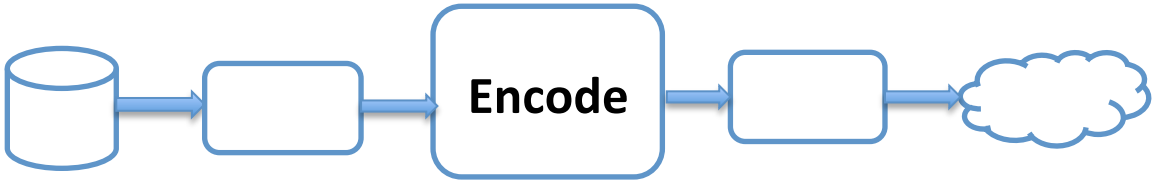
Decode Inspections

- Full decode of source, no errors allowed
- Decoded picture analysis
  - Telecine detection
  - Commercial Black detection
  - Crop detection
- Audio decode: drop out, channel map, etc.

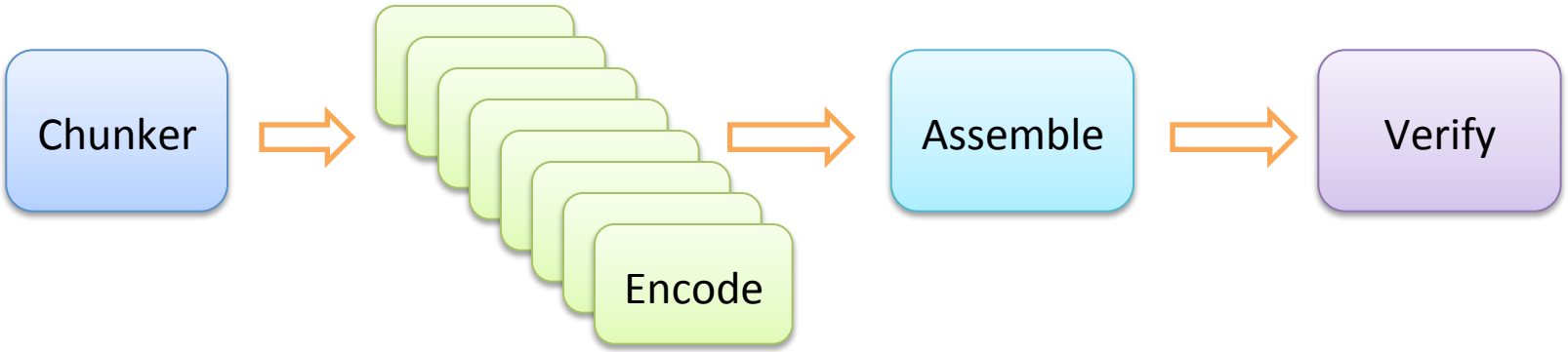


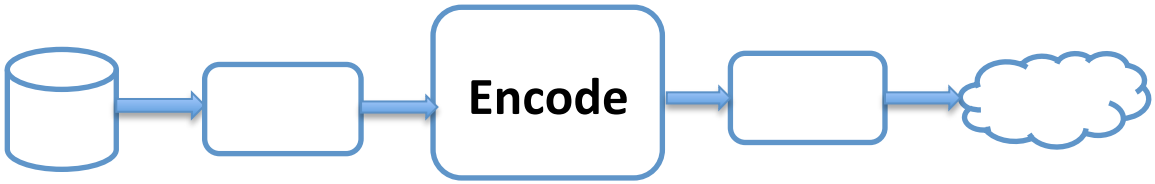
Multiple Sources of Truth

- Each of the major inspections yields A & V sample counts
  - System inspection
  - Decode
  - Index
- All counts must match exactly
- Encoding conformed sources highly predictable



- MAPLE: MAssively ParalleL Encoding
- Video job chunked across multiple EC2 instances
- Progressive model for efficient handling of transient errors
- Post-encode verification detects most encoder issues

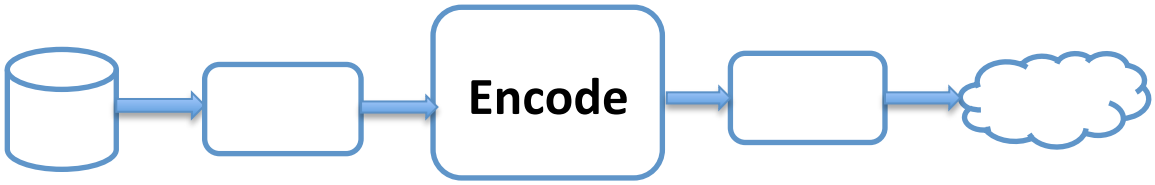




MAPLE in Action: UK Day-after-broadcast

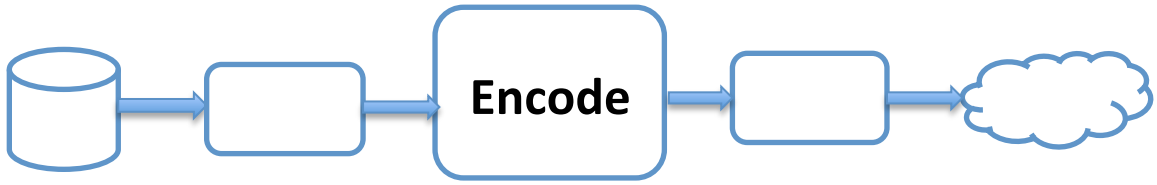
Title	Package ID	Arrived_Date	Inspection Duration Time	Encoding Start Time	Encoding Duration Time (Does not include the waiting time in the queues)	Packaging End Time	Ingest to MPL complete time (hours)
Episode 1	xxxxxx	8/9/13 13:24	3 hours, 11 minutes	8/9/13 17:42	Longest Required ~1 hour Longest - 1 hours 50 minutes	8/9/13 20:47	7.38
Episode 2	xxxxxx	8/12/13 13:58	1 hour, 24 minutes	8/12/13 15:24	Longest Required ~1 hour 10 min Longest - 3 hours 4 minutes	8/12/13 21:05	7.13

- Average time to ingest and encode all bitrates is ~7 hours (~2.5 hours to encode all profiles)
- Compare to Matrix encode ~46 hours for 5.8Mbps 1080p



MAPLE in Action: Worst-Case Encode

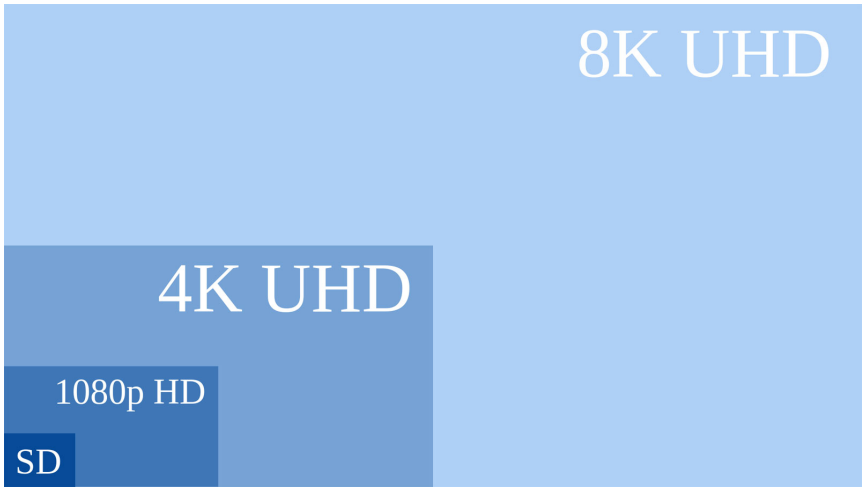
- 4.95 hour title
- 96 hours to encode 1080p in Matrix
- ~900 compute hours before encode completed (due to various transient errors)
- MAPLE encode time ~3 hours.

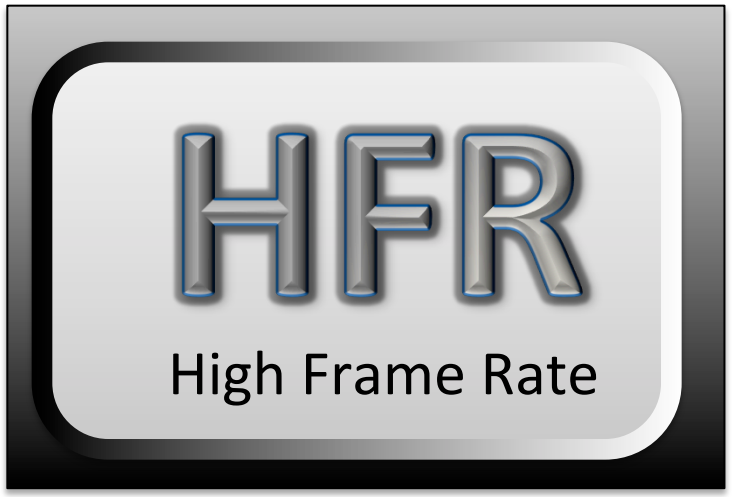


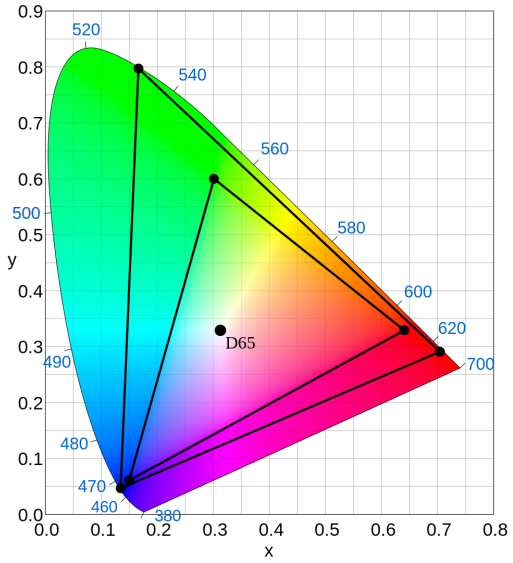
MAPLE Summary

- Able to deliver Day-After-Broadcast at all bitrates
- Tight system tolerances (+/- 1 frame)
- High confidence in quality of encodes
- Progressive model minimizes impact of transient errors
- Still much opportunity to improve performance
- Conclusion: MAPLE is sweet!

Ultra HD 4K


8K UHD
4K UHD
1080p HD
SD


HFR
High Frame Rate


520
540
560
580
600
620
700
490
500
480
470
460
380
D65
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
y
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
x

Ultra HD 

Challenges

- Emerging standards (HEVC, HDR, etc.)
- Bitstream analyzer immaturity
- Few decoders
- Large Source files (DPX ~2TB/hour)

Ultra HD 

Working with 2TB file

- Too big to land on one EC2 drive for inspection
- Downloading could take 24 hours
- Difficulty dealing with transient errors
- Solution: Chunked inspections

Ultra HD 4K

Chunked Inspections

- Able to manage large files efficiently
- Parallel inspections run faster
- Progressive model to minimize transient errors
- Initially DPX, now includes IMF/MXF and ProRes
- Ingest and encode DPX and IMF/MXF at scale

Source Format Roadmap

Supported Formats

- MPEG2 Transport Stream
- Apple ProRes
- DPX
- IMF/MXF (new)

Source Format Roadmap

M2TS

- Well supported but…
- Not really a file format
- Serial inspections
- Video quality not great

Source Format Roadmap

ProRes

- High Quality
- Parallel & Progressive processing
- But … Proprietary format

Source Format Roadmap

DPX

- Very High Quality
- Parallel & Progressive processing
- But … really big
- ~2TB/hour
- S3 5TB Object size limit
- Not practical for scale

Source Format Roadmap

IMF/MXF (JPEG 2000)

- Very High Quality
- Parallel & Progressive processing
- Emerging standard
- Ability to deliver incremental revisions, localized playlists, etc.
- First 4K IMF/MXF assets ingested in Feb. 2014

Source Format Roadmap

Summary

- MPEG2 TS is legacy and not suitable for UHD
- ProRes is proprietary
- DPX is unwieldy
- IMF/MXF is the future
  - Netflix heavily investing in IMF
  - Currently the preferred source format
  - Eventually, may be required format

HEVC Thoughts

- Evaluated many HEVC encoders, currently we see
- No efficiency gain vs. x264 (non-realtime, <=1080p)
- Quality about same as x264
- Much longer encode time (~10x)
- But … very early in HEVC lifecycle
- In ~2 years, we expect
  - 20%-30% encoding efficiency vs. x264
  - Encode time near x264
- We are very excited about HEVC
- Most all new dev is in 4K/HEVC

Random Trivia

- Reloaded transition: every AAC & DD asset was re-encoded …
- 50+ encodes per video (codecs, packaging, bit rates)
- Currently C1M, no GPU acceleration
- 4K DPX workflow stood up in 2 months
- Initial IMF/MXF workflow stood up in 2 months

Questions?

David Ronca

Manager, Encoding Technology

dronca@netflix.com