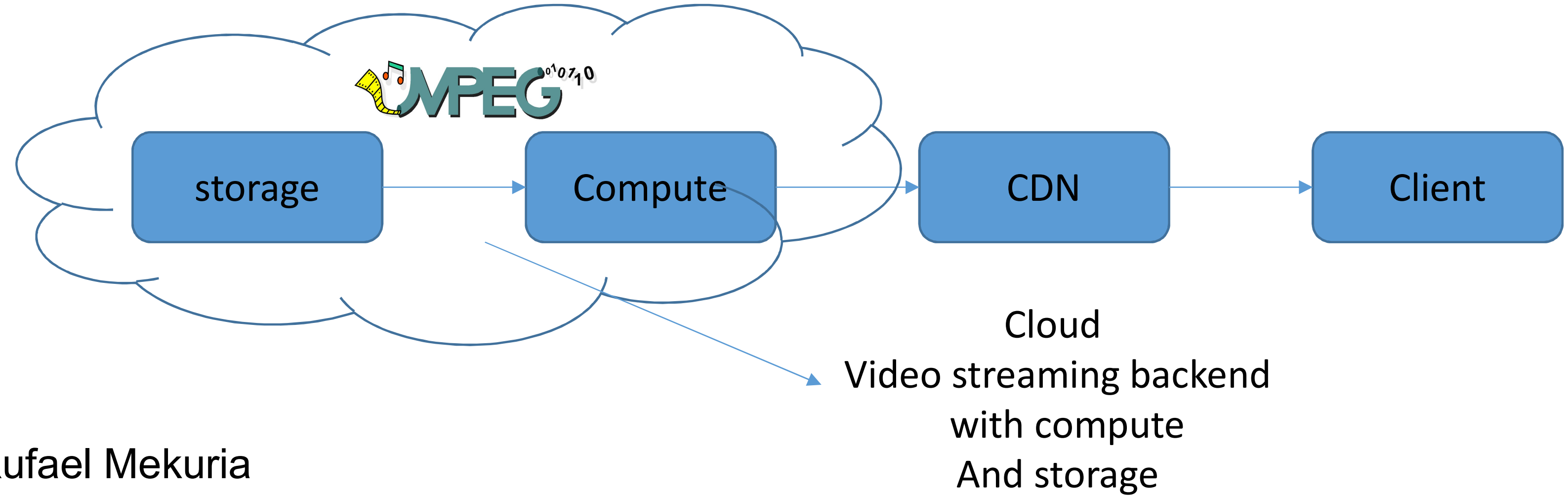
Unified Streaming
Platform

Performance Assessment and Improvement of the video streaming backend with cloud storage and On-the-Fly format conversion



Rufael Mekuria
Christina Kylili
Arjen Wagenaar
Dirk Griffioen

Packet Video Workshop, Amsterdam, June 12 2018

Overview

Unified Streaming
Platform

About Unified Streaming

Cloud Streaming backend with storage and compute

Performance assessment

Improvement with MPEG-4 dref approach

Performance assessment with dref approach

Large scale testing (bonus)

Conclusion, future work and standards

http://docs.unified-streaming.com/documentation/vod/optimizing-storage-caching.html

http://docs.unified-streaming.com/documentation/vod/optimizing-storage-caching.html

Unified Streaming

Unified Streaming Platform

Software for video streaming workflows

DRM, Packaging, Content Stitching, live video

Embedded in cloud, Telco and CDN environments

Standards: DASH-IF, MPEG, 3GPP, DVB

Amsterdam, NL

Unified Streaming B.V.
Overtoom 60 - 3
1054 HK Amsterdam

Burbank, USA

Unified Streaming Inc.
2600 W. Olive Avenue, Suite 500
Burbank CA, 91505 USA

Cloud: origin server and packager Azure, Amazon and soon Google Market place

http://docs.unified-streaming.com/documentation/vod/optimizing-storage-caching.html

Cloud Streaming backend with storage and compute

Unified Streaming Platform

Modern video streaming uses cloud infrastructure which goes beyond the simple client server e.g. Netflix, BBC iPlayer, HBO, ViaPlay etc..

Compute and storage available as separate resources in cloud infrastructure

Combining storage and compute key for efficient video streaming (both quality and cost wise).

Commercial video platforms use this, but implementation details often hidden

We deliver software for video streaming platforms, we help you design/optimize your backend, so we Are open about the video streaming backend.

By doing so we identify bottlenecks and research challenges for video streaming in current architectures

http://docs.unified-streaming.com/documentation/vod/optimizing-storage-caching.html

Cloud Streaming backend with storage and compute

Object Storage: ideal for storing large asset repositories for VoD/DVR due to persistence HTTP interface etc, flexibility: e.g amazon s3 Openstack swift, low costs

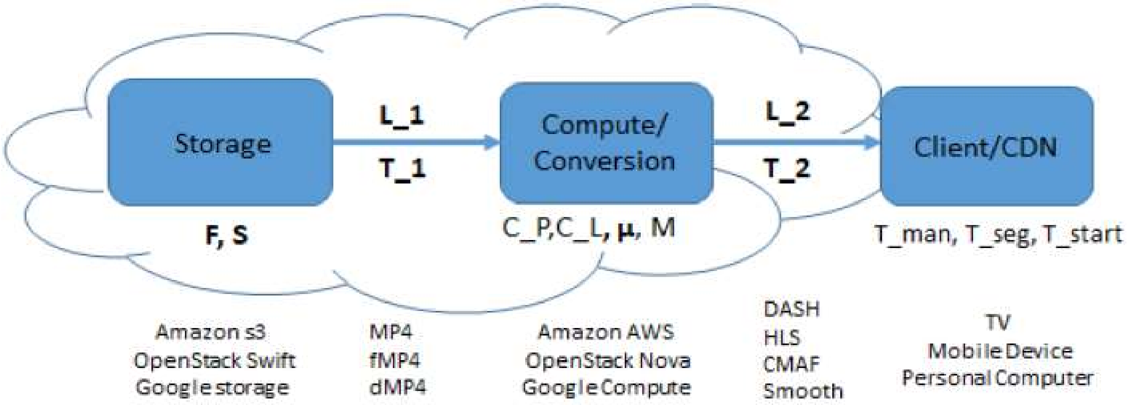
Compute capabilities: virtual machine, container, deal for conversion e.g. transcoding, format conversion, manifest generation, personalization of presentation, reduces redundant storage, Time to market etc.

***Combining storage and compute* in cloud key** for efficient video streaming

We provide the ***performance analysis*** and propose an improvement by ***introducing a new intermediate file format*** -> large increase in **throughput/conversion efficiency at compute node was achieved**
Reduced startup delay was achieved

Performance Assessment: KPI

Unified Streaming Platform

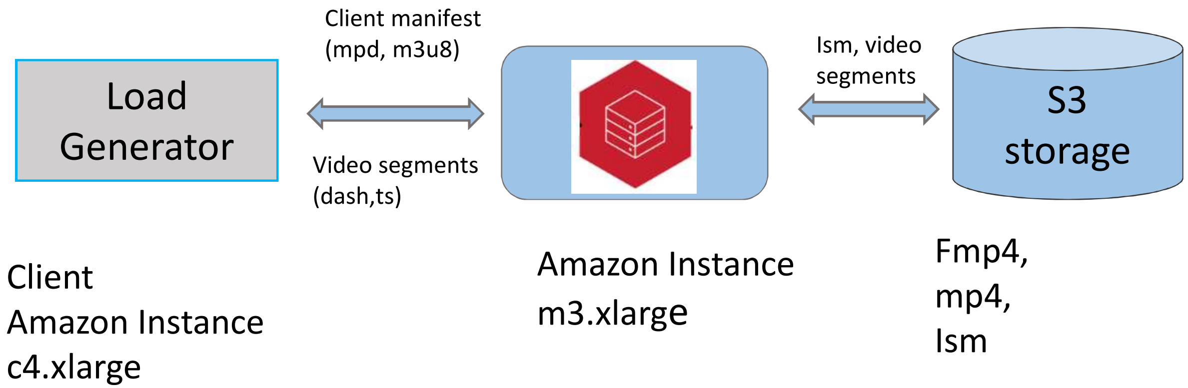


Symbol	*Description*
F	Format{fragmented, non-fragmented, d-ref}
S	Storage space used at the cloud storage
L_1	Latency from storage to compute node
T_1	Incoming traffic to compute node from storage
C_P, C_L	Compute power usage, compute latency
μ	Conversion efficiency of the node (T_2/T_1)
M	Node memory usage (including caching)
N_S	Number of request to storage
L2	Latency between node and client
T_2	Outgoing traffic volume from compute node to clients
T_seg	Time to request a segment
T_man	Time to request a manifest file
T_start	Startup delay to start playing a video with a player

1. Latency (ms), including startup delay, segment delay
2. Data volume throughput
   Tensor (MB/s received and send from server)
   AB (request /second)
3. Conversion efficiency of media processing server

Performance Assessment: Setup

Unified Streaming
Platform



Load Generator tools: Tensor and AB → different testing range

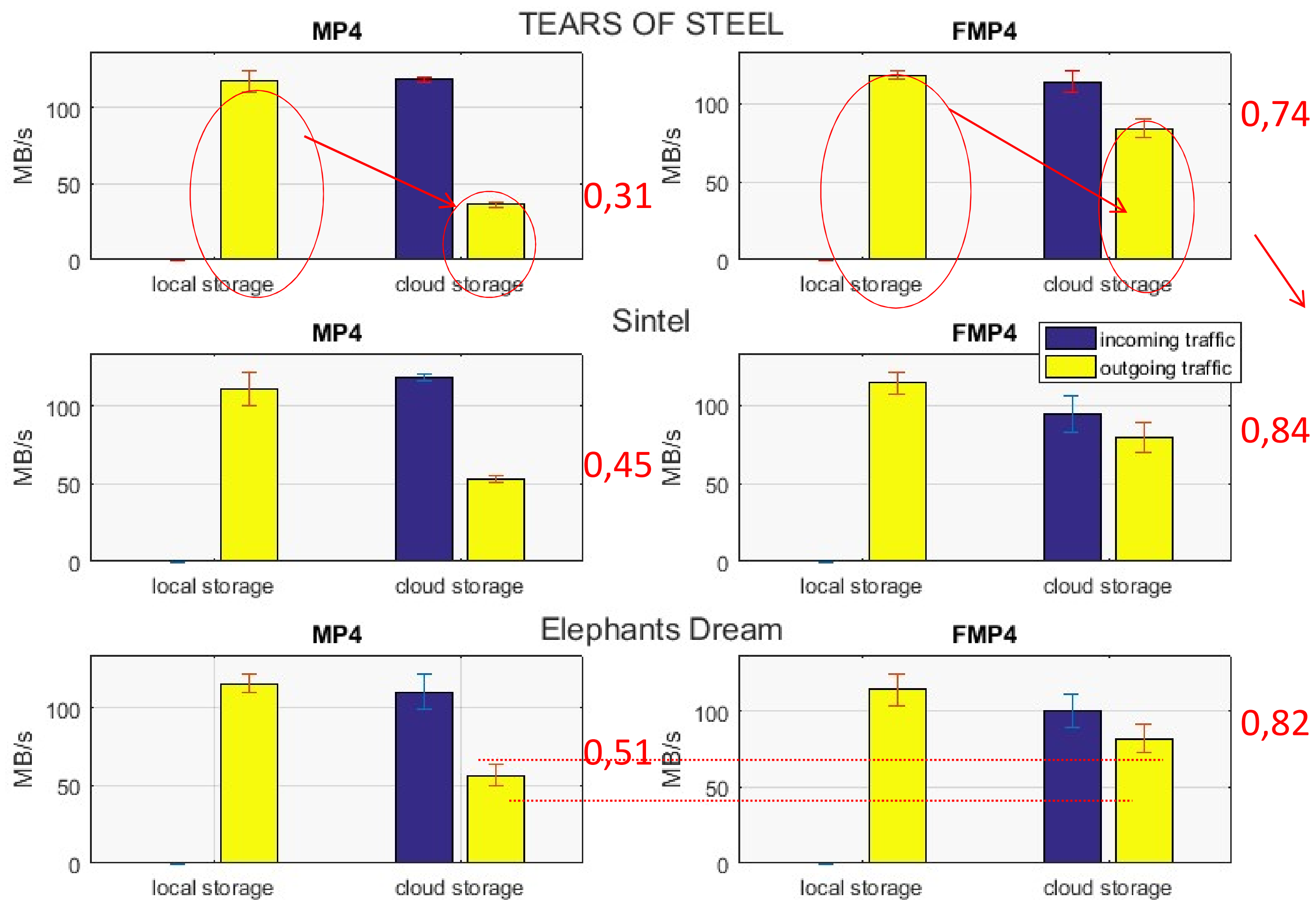
Unified Origin: dynamic packaging and manifest generation: stream DASH+HLS

Simple Storage Solution (S3) as object based storage

3 movies packaged with Unified Packager in MP4 and fMP4

Performance Assessment: Result

Unified Streaming Platform



The Conversion node needs more data than what it produces

$$\frac{\text{Dout}}{\text{Din}}$$

Performance Assessment: Analysis

Unified Streaming Platform

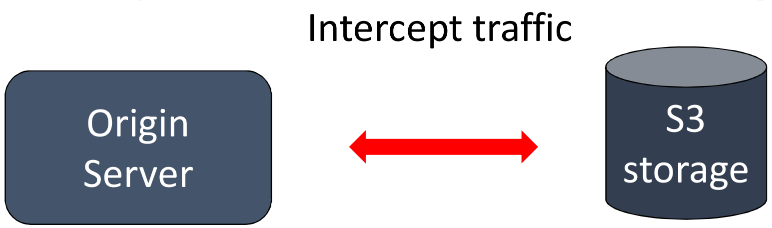
Outgoing traffic decreases with cloud storage

Latency when backend storage is increased→ due to extra communication between origin s3 <20ms

Maximum throughout of the instance cannot be used →Resources go to waste

Traffic Analysis: manifest generation

Unified Streaming Platform



Origin does multiple byte range request to S3

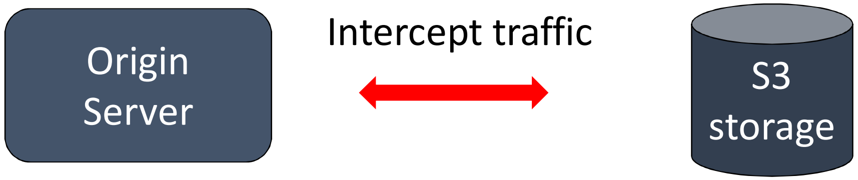
MP4	fMP4
Ism	Ism
ftyp	ftyp
Moov box header	moov (hundreds of bytes)
moov (hundreds of KB)	Mfra size
	Mfra (few KB)
	last moof box header(DASH)
	Last moof (DASH)

To see bitrates

Requests for each bitrate stream to construct URL segments

Traffic Analysis: segment generation

Unified Streaming Platform



MP4	fMP4
Ism	ism
ftyp	ftyp
mvhd	moov (hundreds of bytes)
moov (hundreds of KB)	Mfra size
	Mfra (few KB)
mdat	Moof & mdat box

Locate bitrate stream

Locate samples (indexing and timing info)

Media samples of segment

So..

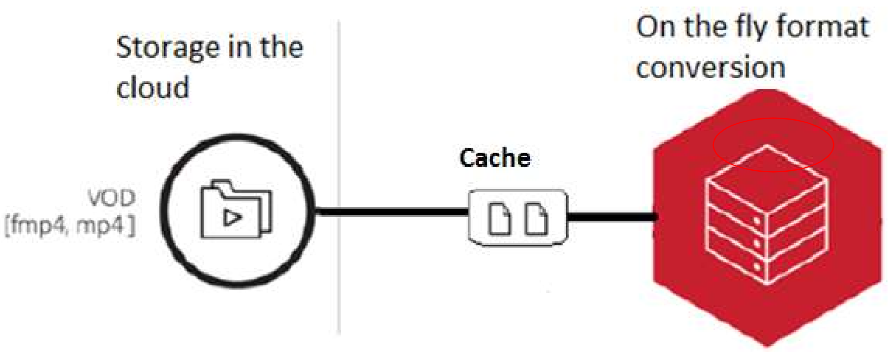
- Critical data for on-the-fly conversion: sample boxes
  - In mp4 this is bigger than fmp4 → bad performance
  - Optimal file format could be designed

- Caching the critical data closer to the conversion server can improve communication

Improvement Proposal

- Cache the critical data close to server:
  - metadata & ism

- Reduce the requests to S3:
  - Only for media samples



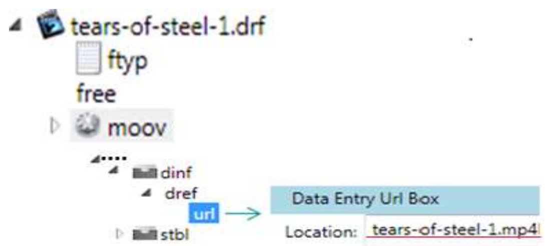
- Using existing technology based on ISOBMFF

DREF MPEG-4

- Part of ISOBMFF media file specification

- Only moov box with metadata, no media data

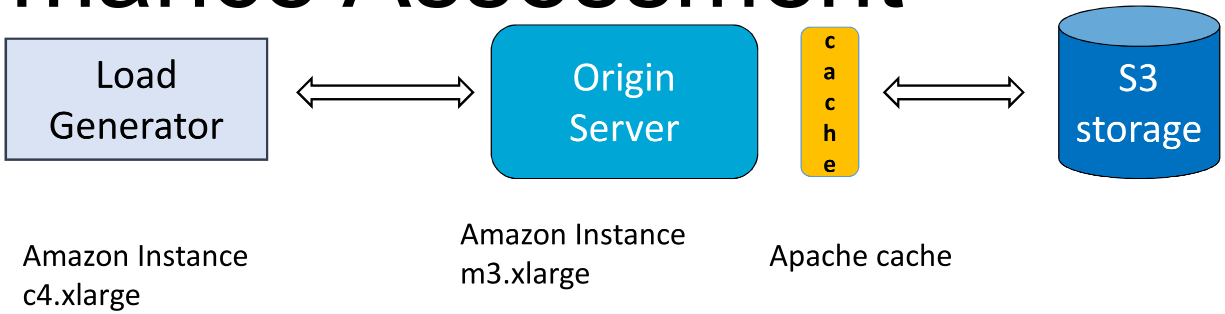
- Points to external file of the media data



- Same structure for fMP4 and MP4

Unified Streaming Platform

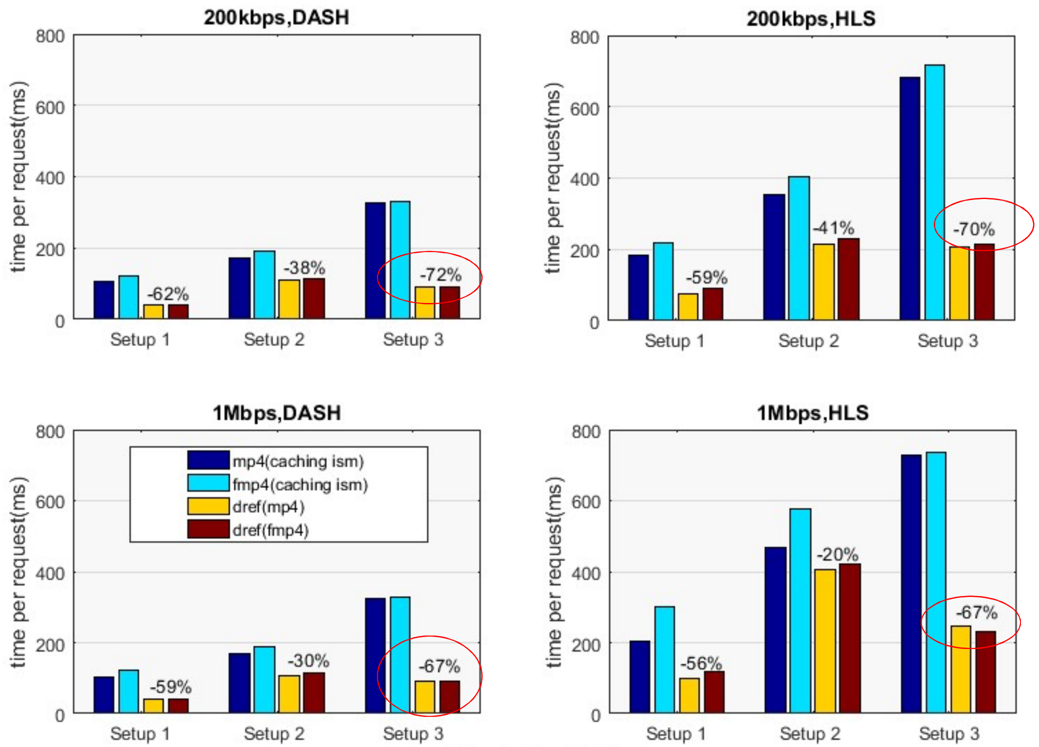
Performance Assessment



- AB for understanding the effect of caching
  - Requesting a manifest
  - Requesting a segment

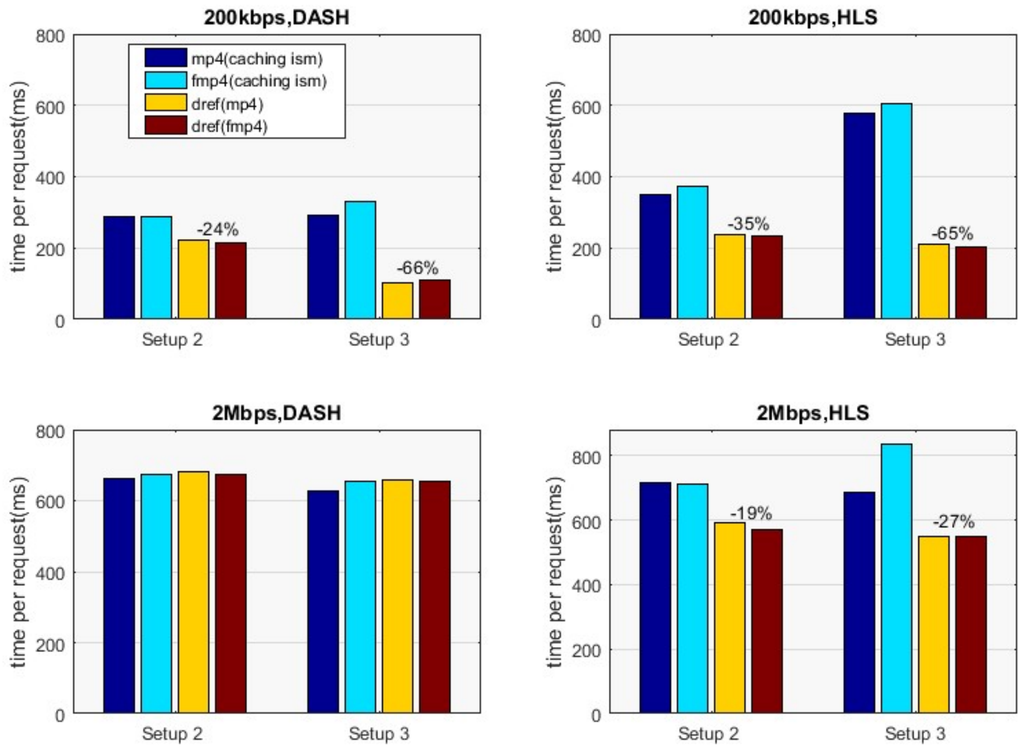
- Tested configurations
  - Setup 1: Client, Origin, Storage in the same cloud environment.
  - Setup 2: Origin, Storage in one cloud. Client in a different cloud.
  - Setup 3: Origin and Client in one cloud. Storage in a different cloud.

Results: Requesting a Segment (1)



Sintel, 480p

Results: Requesting a Segment (2)

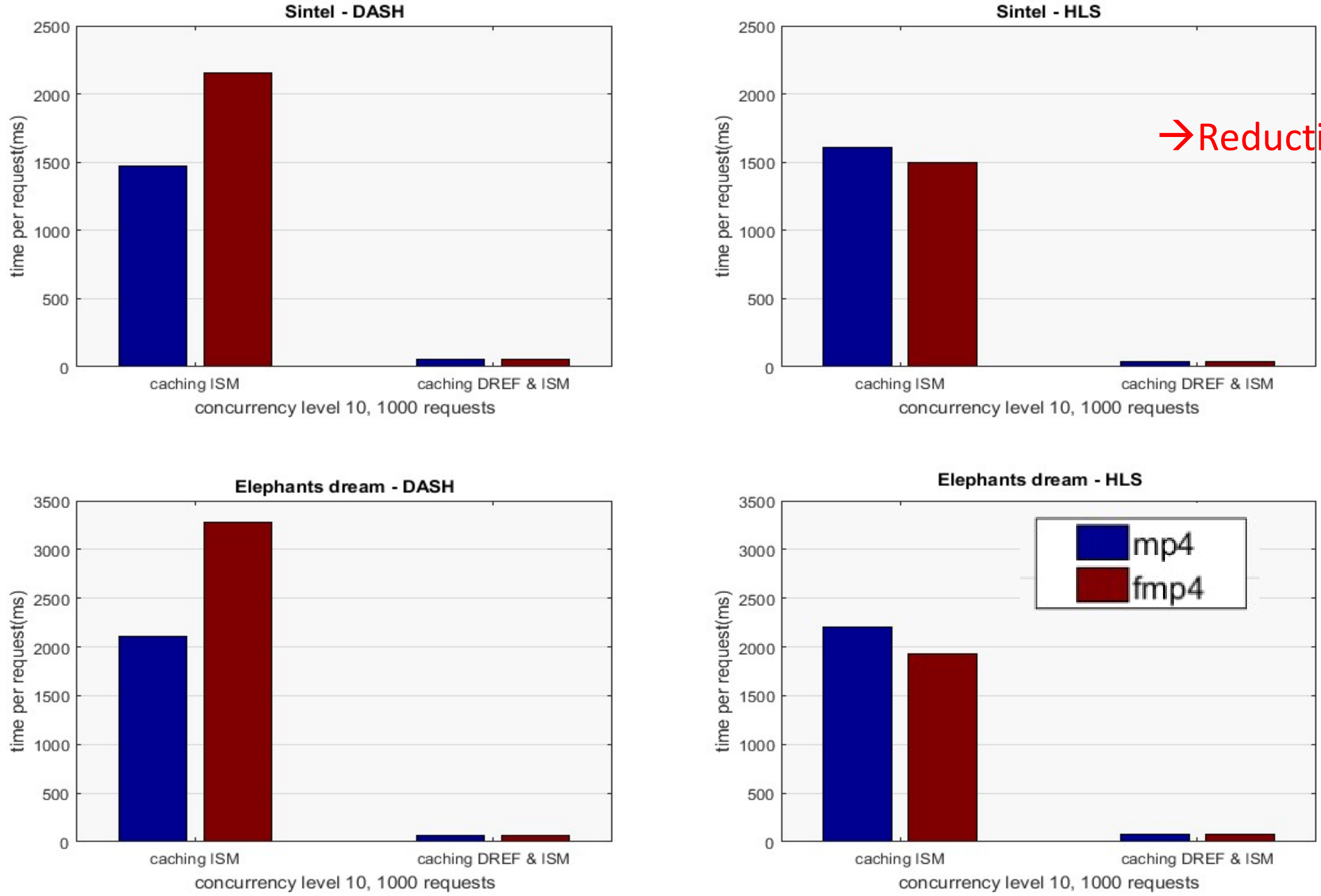


Elephants Dream, 720p

http://docs.unified-streaming.com/documentation/vod/optimizing-storage-caching.html

Unified Streaming Platform

Results: Requesting a manifest file



→Reduction: 97%

→Could decrease startup delay

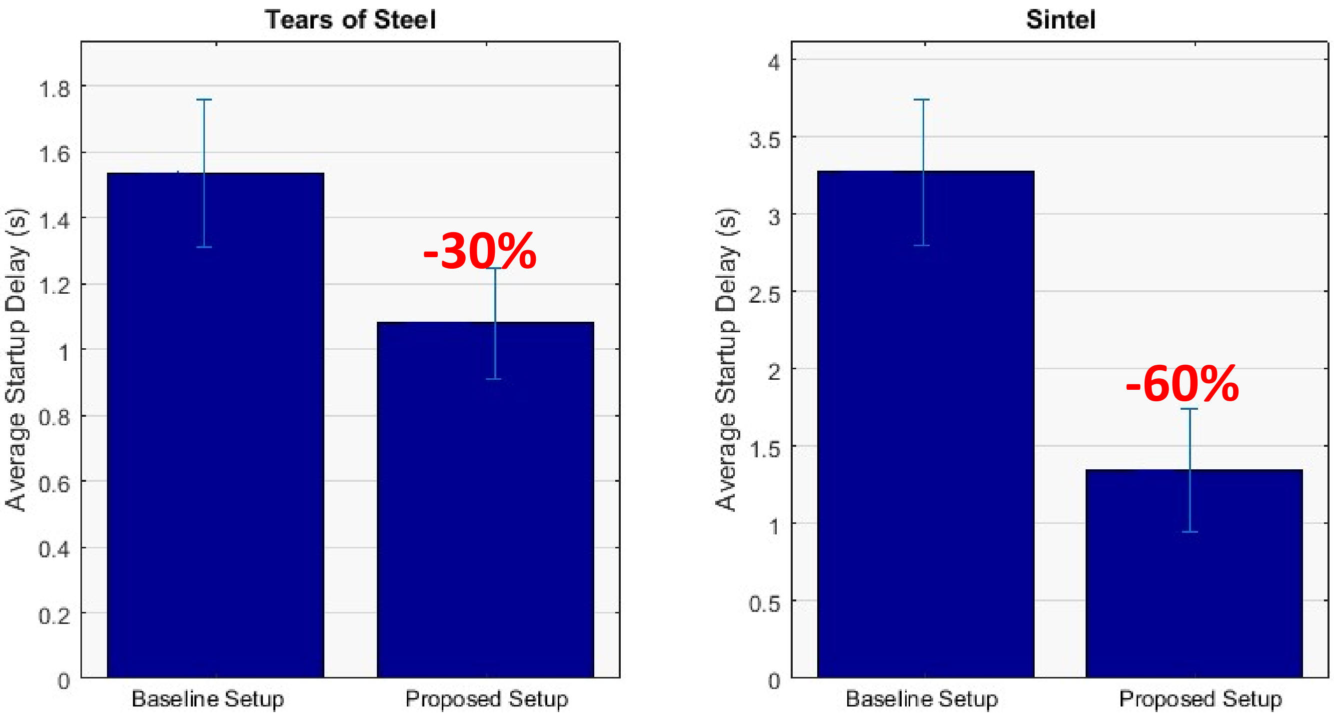
Measuring startup delay

- Time from when the viewer intends the video to play until the first frame of the video is displayed
- Measure the time manually:
    - Dash.js 2.5.0 player
    - Using screen recorder
    - Measure time in slow speed
    - Experiment repeated 10 times
    - Sintel, tears of steel, mp4 case
    - Setup 3 with remote storage
- Compare startup delay between baseline and proposed solution

http://docs.unified-streaming.com/documentation/vod/optimizing-storage-caching.html

Measuring startup delay

- Average Startup Delay



http://docs.unified-streaming.com/documentation/vod/optimizing-storage-caching.html

Conclusions

- Caching the dref performs at least as good as non caching and even better
- Startup delay is decreased
- Less performance variation between MP4 and fMP4
- What about large scale testing? (bonus)

http://docs.unified-streaming.com/documentation/vod/optimizing-storage-caching.html

Unified Streaming Platform

Large Scale testing

- Realistic workload from video players → Tensor
- High concurrent video traffic
- Requires to Tune Apache:
  - High concurrency →multiple activities executed at the same time: cache connections, origin connections.
  - Heavy load on server → efficient scheduling of connections
  - See paper appendix for tuned configuration

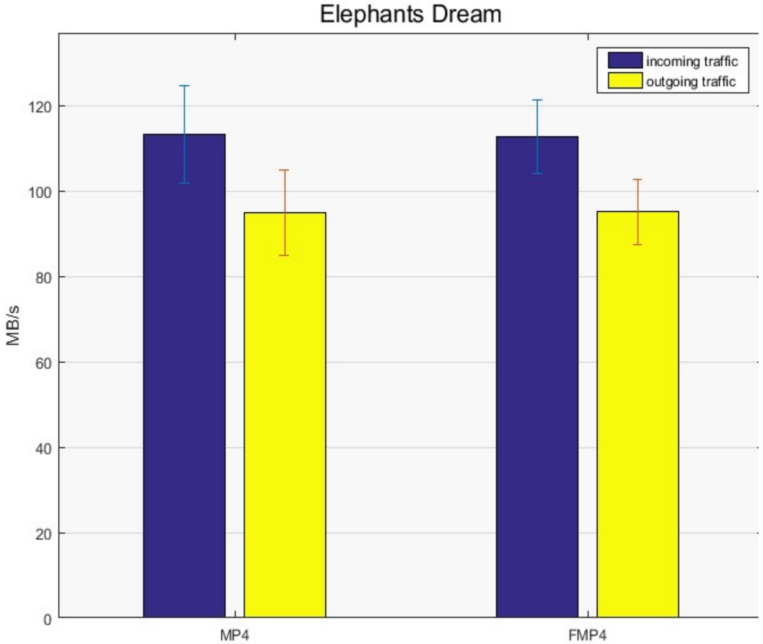
C10K problem: Hardware is not an issue, Software implementations (OS Multi threading, context switching) can be a bottleneck Use suitable concurrency models ,I/O strategies offered by servers

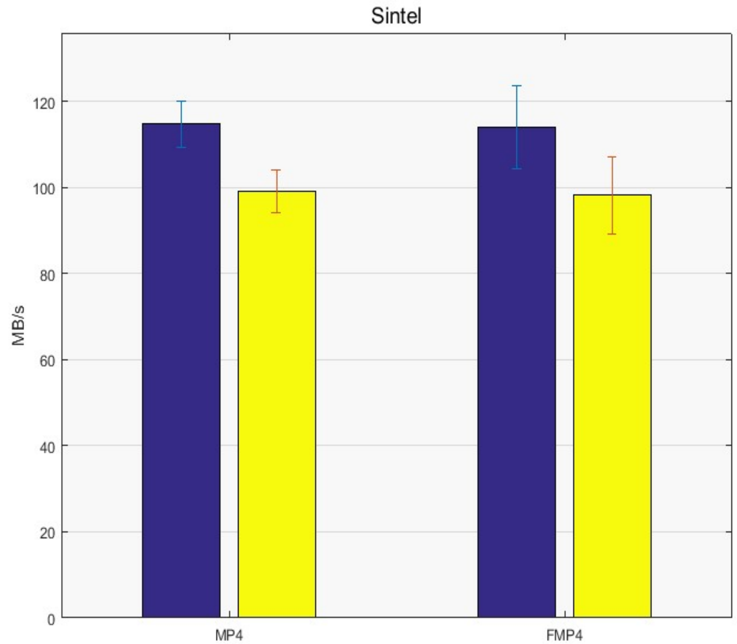
Unified Streaming Platform

http://docs.unified-streaming.com/documentation/vod/optimizing-storage-caching.html

Unified Streaming Platform

Large scale Testing: Results





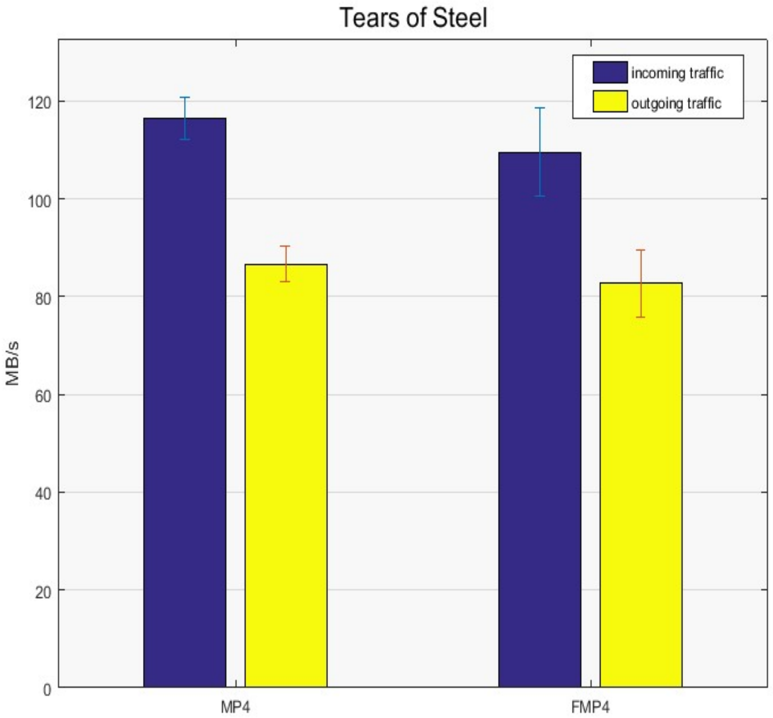
MP4	**0,84**	**65%**
fMP4	0,84	2%

MP4	**0,86**	**91%**
fMP4	0,86	2%

http://docs.unified-streaming.com/documentation/vod/optimizing-storage-caching.html

Unified Streaming Platform

Large scale Testing: Results



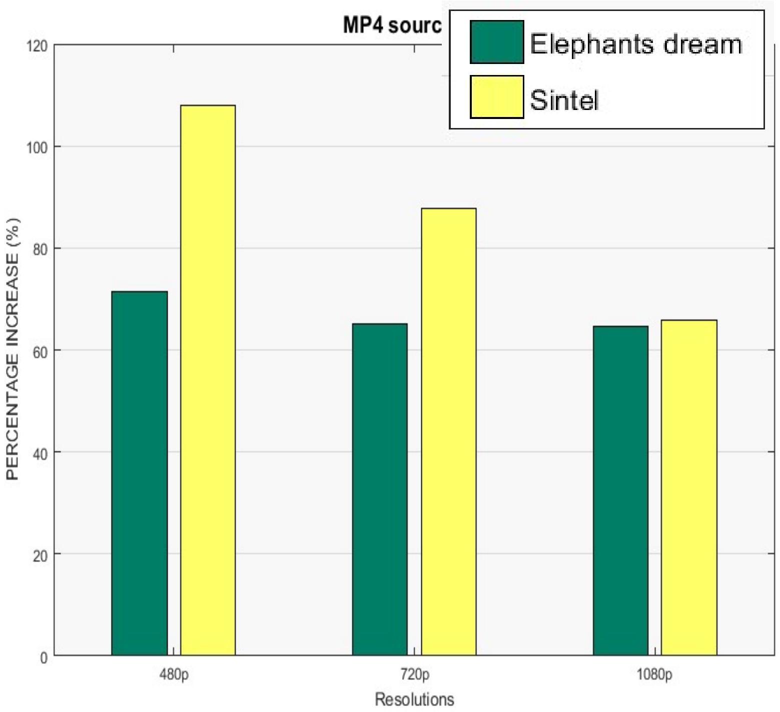
MP4	**0,74**	**139%**
fMP4	0,75	1%

http://docs.unified-streaming.com/documentation/vod/optimizing-storage-caching.html

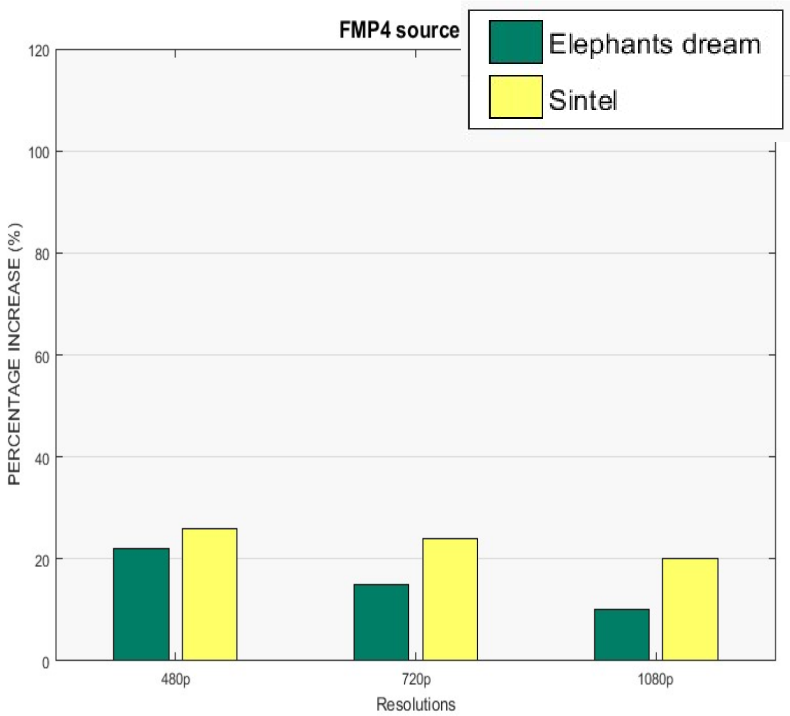
Unified Streaming Platform

Increase in throughput per resolution

MP4 source file



FMP4 source file



http://docs.unified-streaming.com/documentation/vod/optimizing-storage-caching.html

Conclusions

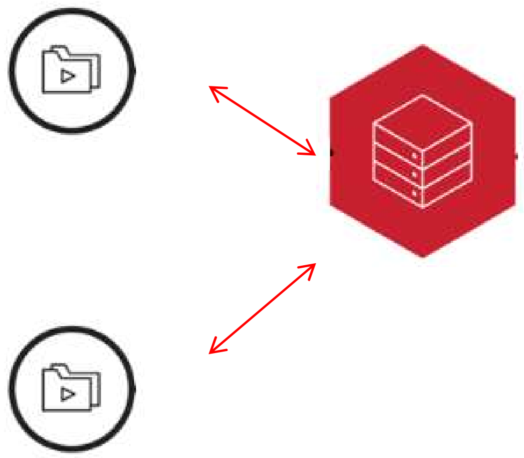
- Increased outgoing traffic towards client
- Conversion efficiency increases for MP4
- Latency is reduced for segment and manifest request
- Design and standardization of optimal formats such as dref for media processing operations can improve streaming performance, target of emerging NBMP standard

Future work

- Using a different media processing function
  - Stitching content from multiple sources
    - Crowd-sourcing
    - Ad insertion
    - Personalize streams
- Setup is reproducible
- Trial license for conversion software
- Paper on Thursday on profiling conversion

Server with machine learning and telemetry

14:45-16.15



Is dref sufficient? What more can we do?

- Unified Streaming docs:
- http://docs.unified-streaming.com/documentation/vod/optimizing-storage-caching.html